



Universidad
Carlos III de Madrid

Departamento de Ingeniería del Software.

PROYECTO FIN DE CARRERA

Análisis de los procesos de verificación y validación en las organizaciones software.

Autor: Jorge Zamora Hernández.

Tutor: Ana Sanz Esteban.

Leganés, Febrero de 2011

Título: Análisis de los procesos de verificación y validación en las organizaciones software.

Autor: Jorge Zamora Hernández.

Director: Ana Sanz Esteban.

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Agradezco a Alberto García y Francisco Valera de Ingeniería Telemática la plantilla inicial de la que esta ha sido adaptada.

Llegado el momento de dar por finalizado este proyecto y con él mis estudios en la universidad Carlos III, son muchas las personas que se me vienen a la mente a la hora de recapitular los largos años que me han llevado esta andadura.

En primer lugar, no paro de dar las gracias por la familia que tengo, ellos son los culpables de que cada día el llegar a casa sea un verdadero descanso y son quienes han tenido que aguantar mis malos momentos y desilusiones, dándome siempre el apoyo y los ánimos necesarios para poder conseguir todo lo quiero, a vosotros os lo debo todo. Gracias mamá, papá y Alber.

Durante todos estos años he tenido la fortuna de contar con la compañía de un nutrido grupo de individuos que habitualmente suelo llamar amigos. La verdad es que hablar de todos vosotros me daría para escribir otro proyecto, y aun así creo que me quedaría corto. Siempre me habéis llenado de energía, aunque sé que a veces tanta sea difícil de soportar y he compartido toda mi vida con vosotros: risas, confesiones, cotilleos, cervezas, la expedición del pompón, viajes, snow, tardes de biblioteca, conciertos. Gracias por todo Miry, Tormentas, Mariete, Falute, Winni, Aberto, Dancer, Miguelín, Tania y por supuesto a todos “los changos”.

Agradecer también a mis compañeros de universidad los ratos tanto dentro como fuera de la universidad que han supuesto para mí el recuerdo más agradable que me pueda llevar.

Por último, dar las gracias a mi tutora, Ana Sanz, por su esfuerzo y dedicación a la hora de ayudarme a desarrollar este proyecto.

Resumen

La verificación y validación es una actividad que juega un papel importante en la consecución de procesos y productos de calidad. Por ello y dado los problemas que históricamente han surgido en los desarrollos software, su estudio y mejora supone un avance importante para la futura evolución de las tecnologías de la información.

En el presente documento se establece la evolución de dicha actividad y su actual impacto en los distintos tipos de organizaciones software que las implementan, con el fin de poder extraer algunos de los conceptos más importantes en su implantación y desarrollo, y así elaborar una guía de buenas prácticas que permita seleccionar las mejores opciones para realizar las pruebas de software en función de las características de una organización.

Para lo cual se presentan algunos conceptos que influyen directamente en el desarrollo de la verificación y validación software como son: los modelos de calidad, los modelos de mejora del proceso de pruebas, las estructuras organizativas, las competencias y los perfiles profesionales.

Palabras clave: calidad, pruebas, v&v, TMap, TPI, TMMI, CMMI, TMM, verificación, validación, outsourcing, SQA

Abstract

Verification and validation is an activity that plays an important role in the achieving of quality processes and products. Therefore, and given the problems that have historically emerged in software development, its study and improvement is a significant advance for the future development of information technology.

This document provides the evolution of current activity and its impact on different types of software organizations that implement them, in order to extract some of the most important concepts in its implementation and development, so we can develop a guide of good practice in order to select the best options for software testing based on the characteristics of an organization.

For which they are some concepts that directly influence the development of software verification and validation including: models of quality improvement models in the testing process, organizational structures, competencies and professional profiles.

Keywords: quality, testing, v & v, TMap, TPI, TMMI, CMMI, TMM, verification, validation, outsourcing, SQA.

Índice general

1. INTRODUCCIÓN Y OBJETIVOS	1
1.1 Introducción	1
1.2 Objetivos	4
1.3 Fases del desarrollo	4
1.4 Medios empleados.....	5
1.5 Estructura de la memoria.....	5
2. ESTADO DEL ARTE.....	6
2.1 Antecedentes	6
2.2 La calidad del software	12
2.3 El proceso software	14
2.3.1 <i>El ciclo de vida</i>	15
2.4 El proceso de verificación y validación	18
2.4.1 <i>Técnicas de verificación y validación</i>	20
2.5 Las pruebas software.....	25
2.5.1 <i>Los principios de las pruebas</i>	27
2.6 La mejora de los procesos software	27
2.6.1 <i>Modelos de mejora de los procesos software</i>	28
2.6.2 <i>Modelos de mejora de los procesos software</i>	30
2.7 Estructura organizacional	32
2.8 Competencias y perfiles profesionales	34
2.9 Resumen del estado del arte	36
3. PROPUESTA	38
3.1 Fase de búsqueda de información	38
3.2 Fase de análisis de la información.....	42
3.2.1 <i>CMMI: Modelo de capacidad y madurez integrado</i>	46
3.2.2 <i>TMM: Modelo de madurez de las pruebas</i>	59
3.2.3 <i>TMMI: Modelo de madurez de las pruebas integrado</i>	68
3.2.4 <i>TPI: Modelo de mejora del proceso de pruebas</i>	82
3.2.5 <i>TMap Next: Aproximación a la gestión de pruebas</i>	91
3.2.6 <i>La organización de las pruebas convencional</i>	107

3.2.7 Los grupos SQA	108
3.2.8 Outsourcing.....	118
3.2.9 Los grupos de pruebas	134
3.2.10 Perfiles profesionales de las TI.....	136
3.2.11 Competencias dentro de las TI.....	139
3.2.12 La verificación y la validación bajo el marco de las TI	143
3.3 Fase de obtención de resultados	160
4. CONCLUSIÓN Y FUTURAS LÍNEAS	169
4.1 Conclusión.....	169
4.2 Futuras líneas	170
5. PRESUPUESTO	172
5.1 Resumen y diagrama de Gantt	172
6. GLOSARIO	177
7. REFERENCIAS.....	178

Índice de figuras

<i>Figura 1. Modelo en cascada del ciclo de vida.</i>	16
<i>Figura 2. Modelo-V del ciclo de vida.</i>	17
<i>Figura 3. V&V en el ciclo de vida software.</i>	19
<i>Figura 4. Técnicas de caja negra y blanca.</i>	23
<i>Figura 5. Actividades de un proceso de mejora.</i>	29
<i>Figura 6. Esquema de la estructura de un perfil profesional.</i>	35
<i>Figura 7. Estructura de los componentes de CMMI.</i>	47
<i>Figura 8. Estructura de la representación por etapas de CMMI.</i>	49
<i>Figura 9. Estructura de la representación continua de CMMI.</i>	49
<i>Figura 10. Estructura de TMM.</i>	60
<i>Figura 11. Estructura de TMMI.</i>	69
<i>Figura 12. Estructura de la representación por etapas de TMMI.</i>	70
<i>Figura 13. Estructura de TPI.</i>	83
<i>Figura 14. Estructura de TMap Next [DRIEL].</i>	92
<i>Figura 15. Proceso de BDTM.</i>	94
<i>Figura 16. Modelo del ciclo de vida TMap [DRIEL].</i>	96
<i>Figura 17. Niveles de las técnicas de estimación.</i>	103
<i>Figura 18. Ciclo de vida del SQA.</i>	111
<i>Figura 19. Ciclo de vida del outsourcing.</i>	121
<i>Figura 20. Relación entre los tipos de outsourcing.</i>	123
<i>Figura 21. Impacto de la crisis (miembros de la IAOP) [CORB09].</i>	127
<i>Figura 22. Estado actual del outsourcing (miembros de la IAOP) [CORB09].</i>	128
<i>Figura 23. Casos de negocios solicitados miembros de la IAOP [CORB09].</i>	128
<i>Figura 24. Selección de proveedores miembros de la IAOP [CORB09].</i>	129
<i>Figura 25. Estado de las oportunidades de outsourcing (miembros de la IAOP).</i>	129
<i>Figura 26. Estructura de los grupos de pruebas independientes.</i>	135
<i>Figura 27. Estructura del grupo de pruebas dependiente.</i>	136
<i>Figura 28. Propuesta de la estructura de un perfil profesional.</i>	155
<i>Figura 29. Metodología para la implantación de un proceso de v&v.</i>	163

<i>Figura 30. Tareas del estudio del indicie de los errores en el desarrollo.</i>	164
<i>Figura 31. Tareas de la delimitación de los objetivos de calidad de empresa.</i>	164
<i>Figura 32. Tareas del estudio del nivel de madurez de la organización (etapas).</i>	164
<i>Figura 33. Tareas del estudio el nivel de madurez de la organización (continuo).</i>	165
<i>Figura 34. Tareas del estudio del nivel de madurez del proceso de pruebas.</i>	165
<i>Figura 35. Tareas de la selección del nivel de madurez objetivo.</i>	165
<i>Figura 36. Tareas de implantación de las acciones de mejora.</i>	166
<i>Figura 37. Tareas de la selección de una organización de pruebas.</i>	166
<i>Figura 38. Tareas de la planificación de una organización de pruebas.</i>	166
<i>Figura 39. Tareas de la distribución de funciones de la organización de las pruebas.</i>	167
<i>Figura 40. Tareas de la selección de una organización de pruebas.</i>	167
<i>Figura 41. Tareas de la definición de las trayectorias profesionales.</i>	167
<i>Figura 42. Tareas de la implementación de la organización de las pruebas.</i>	168
<i>Figura 43. Tareas de la recogida y análisis de los resultados.</i>	168
<i>Figura 44. Diagrama de Gantt.</i>	173
<i>Figura 45. Presupuesto.</i>	174

Índice de tablas

<i>Tabla 1. Términos de búsqueda.</i>	41
<i>Tabla 2. Tabla de relación entre los niveles CMMI.</i>	50
<i>Tabla 3. Tabla de relación de niveles de madurez y APs en CMMi.</i>	52
<i>Tabla 4. Relación entre niveles de madurez y objetivos genéricos.</i>	55
<i>Tabla 5. Tabla de relación de niveles y objetivos de madurez en TMM.</i>	62
<i>Tabla 6. Tabla de relación de niveles de madurez y APs en TMMi.</i>	73
<i>Tabla 7. Relación entre niveles de madurez y objetivos genéricos.</i>	75
<i>Tabla 8. Soporte del nivel 2 de CMMI en el nivel 2 de TMMI.</i>	81
<i>Tabla 9. Soporte del nivel 3 de CMMI en el nivel 2 de TMMI.</i>	81
<i>Tabla 10. Soporte del nivel 2 de CMMI en el nivel 3 de TMMI.</i>	82
<i>Tabla 11. Soporte del nivel 3 de CMMI en el nivel 3 de TMMI.</i>	82
<i>Tabla 12. Matriz de madurez de pruebas de TPI.</i>	84
<i>Tabla 13. Tabla de relación Actividad/Fase del ciclo de vida de Tmap.</i>	98
<i>Tabla 14. Ventajas e inconvenientes del método convencional.</i>	108
<i>Tabla 15. Elementos y actividades de la Fase 1 del SQA.</i>	112
<i>Tabla 16. Elementos y actividades de la Fase 2 del SQA.</i>	113
<i>Tabla 17. Elementos y actividades de la Fase 3 del SQA.</i>	113
<i>Tabla 18. Elementos y actividades de la Fase 4 del SQA.</i>	114
<i>Tabla 19. Elementos y actividades de la Fase 5 del SQA.</i>	114
<i>Tabla 20. Elementos y actividades de la Fase 6 del SQA.</i>	115
<i>Tabla 21. Elementos y actividades de la Fase 7 del SQA.</i>	115
<i>Tabla 22. Elementos y actividades de la Fase 8 del SQA.</i>	115
<i>Tabla 23. Características de SQA y V&V.</i>	116
<i>Tabla 24. Ventajas e inconvenientes de los grupos SQA.</i>	118
<i>Tabla 25. Características del outsourcing.</i>	120
<i>Tabla 26. Ventajas y riesgos en función de la distancia del outsourcing.</i>	124
<i>Tabla 27. Ventajas y riesgos en función del modelo onsite.</i>	124
<i>Tabla 28. Ventajas y riesgos en función del modelo offsite.</i>	125
<i>Tabla 29. Ventajas y riesgos en función del modelo offshore.</i>	126

<i>Tabla 30. Ventajas y riesgos en función del modelo híbrido.</i>	126
<i>Tabla 31. Ventajas e inconvenientes de los grupos de pruebas independientes.</i>	135
<i>Tabla 32. Ventajas e inconvenientes de los grupos de pruebas dependientes.</i>	136
<i>Tabla 33. Objetivos del proceso de pruebas por perfil profesional.</i>	156

Capítulo 1

Introducción y objetivos

1.1 Introducción

El proyecto que se presenta continuación surge con motivo de la finalización de los estudios en Ingeniería Técnica en Informática de Gestión. Bajo la tutela de Ana Sanz se propone realizar un “*análisis de los procesos de verificación y validación en las organizaciones software*”.

El interés que suscita el tema en cuestión, viene dado por su cada vez mayor peso en el desarrollo software en la actualidad. La importancia del concepto de la verificación y validación (v&v) viene siendo reflejada desde hace más de 30 años, de hecho Michael S. Deutsch [DEUT79] en su estudio “*Verification and Validation*” establece que:

“El desarrollo de sistemas de software implica una serie de actividades de producción en las que las posibilidades de que aparezca el fallo humano son enormes. Los errores pueden empezar a darse desde el primer momento del proceso, en el que los objetivos pueden estar especificados de forma errónea o imperfecta, así como en posteriores pasos de diseño y de desarrollo. Debido a la imposibilidad humana de trabajar y comunicar de forma perfecta, el desarrollo de software ha de ir acompañado de una actividad que garantice la calidad.”

Los proyectos de desarrollo software han padecido tradicionalmente problemas de calidad, tanto en el propio proceso de desarrollo como en los productos de entrega. Esta problemática tiene su origen en las habituales desviaciones sobre los plazos y esfuerzos

CAPÍTULO 1: Introducción y objetivos

previstos y en la frecuente aparición de fallos durante la implantación y operación de los productos resultantes.

El primer problema pone de manifiesto una falta de calidad en el proceso de gestión de los proyectos software: cuanto menor es ésta, peor es el grado de adherencia a los plazos y a los esfuerzos previstos. El segundo problema indica la falta de calidad de los productos desarrollados: cuanto menor es ésta, mayor es el número de defectos y, consecuentemente, mayor será el número de fallos que aparecerán durante la ejecución del software.

Para hacer frente a esta situación la comunidad involucrada en el desarrollo del software ha reaccionado con diversas iniciativas metodológicas, tales como:

- La definición de modelos de referencia, de esta forma nació el Modelo CMMI, diseñado por el SEI, que establece cinco niveles de madurez, especificando para cada uno de ellos los objetivos que deben ser cubiertos para que una organización pueda ser calificada con el nivel de madurez correspondiente. Se calcula que el 75% de las organizaciones dedicadas al desarrollo de software está en el nivel de madurez 1, es decir, en el más bajo.
- El establecimiento de normas y guías para el desarrollo de software. Éstas han sido promovidas o generadas por entidades de reconocido prestigio (IEEE, CEI, BSI, AENOR) y se han orientado a definir el ciclo de vida del software, a normalizar la terminología o a desarrollar aspectos específicos del ciclo de vida, tales como la documentación, la verificación y validación o las pruebas de componentes.

Como resultado de estas iniciativas, las empresas han ido tomando conciencia de la necesidad de mejorar los procesos de desarrollo software demandando nuevos servicios a las empresas consultoras de TI (Tecnologías de la Información).

Sin embargo, la fase de pruebas, tipificada en el ciclo de vida del desarrollo, ha sido a menudo descuidada y, casi siempre, sacrificada ante las presiones sobre el plazo o el coste de los proyectos. Así, frecuentemente se ha incurrido en una planificación deficiente de las pruebas, se ha reducido el alcance de éstas a un mínimo y no se ha realizado una documentación adecuada de todo el proceso. Como consecuencia, muchas de estas pruebas han tenido lugar como parte de la implantación, manifestándose, como fallos, los defectos existentes en el software, y cuya resolución en estas fases tardías ha producido un incremento sustancial del plazo y del esfuerzo respecto a los inicialmente previstos para el proyecto, y, en muchos casos, ha supuesto un efecto negativo sobre la imagen y la confianza del servicio.

La mejora de esta situación ha supuesto un nuevo enfoque basado en la anticipación de la detección de los errores, ya que éstos se producen con mayor frecuencia en las primeras fases del ciclo. Más del 80% de los errores cometidos en la producción de software provienen de las primeras fases del ciclo de vida (análisis de requisitos y diseño funcional y técnico). Por otra parte, el coste de corregir los errores crece exponencialmente según avanza el proyecto, siendo éste significativamente menor cuanto antes son detectados los fallos.

Las principales características de este enfoque se resumen en:

- Las pruebas no son una fase aislada del proyecto sino que constituyen en sí un proyecto con su propio ciclo de vida.
- Los equipos de desarrollo y de pruebas son independientes, con funciones y perfiles diferenciados.
- Antes de la ejecución de las pruebas se lleva a cabo todo un proceso metodológico que facilita y asegura el éxito de las mismas.
- En el momento de la ejecución, está todo previsto, tanto los aspectos funcionales como técnicos, evitándose la improvisación.

Basados en la aplicación de éstas características, a lo largo del proyecto se recurre a una serie de conceptos específicos en el ámbito de las pruebas, con el afán de que éstas queden representadas a la hora de implementar un proceso de pruebas. De esta forma, debido a que las pruebas son consideradas como un proceso en sí mismas, estableciendo su propio ciclo de vida, se hace necesario para su implementación una serie de modelos que guíen su desarrollo, a modo de completar un proceso de calidad. Éstos son los modelos de madurez y de mejora. Además, dada la necesidad de establecer equipos independientes y el establecimiento de las labores de pruebas como funciones especializadas, es necesario establecer las distintas maneras en las que es posible dividir el trabajo dentro de una organización para alcanzar la coordinación de las pruebas, y satisfacer así las necesidades planteadas durante su desarrollo. Para ello se crean y se relacionan distintos grupos orientados al desarrollo de las pruebas dentro de una organización, los cuales cumplirán con las distintas funciones que acarrea la realización de este tipo de actividades. Dichas funciones, además, marcan la composición de un grupo de trabajo, pero para poder ser llevadas a cabo es necesario contar con un personal especializado y con la formación necesaria al respecto. Las características y las habilidades que presenta el personal de una organización son conocidas como las competencias y los perfiles profesionales, da tal forma que marcan las atribuciones y las capacidades que desarrollarán los empleados en sus puestos de trabajo.

Los beneficios de aplicar un proceso de pruebas, formalmente conocido como verificación y validación, no sólo repercute en el área inmediata del desarrollo de los proyectos, sino que se extienden al resto de la empresa, a las distintas áreas de negocio e incluso al cliente final, ya que el departamento de informática al cumplir con su compromisos de plazos, costes y calidad facilita la transición del desarrollo a la producción, ya que los proyectos pasan al entorno de producción con estabilidad, eliminándose la “etapa” de post-implantación, evitando el efecto cascada en el correctivo, y ofreciendo una mejor respuesta a los aspectos técnicos relevantes del ciclo de vida en producción. Todo ello hace que se incremente la confianza y el nivel de satisfacción de los usuarios finales y de los responsables de la explotación del nuevo sistema.

Por último y dada la importancia que se ha puesto de manifiesto por alcanzar mayores niveles de eficiencia en el desarrollo de software, es de considerar un planteamiento más extenso y profundo de lo que habitualmente se conoce como fase de pruebas, ya que aunque no garantice por sí misma la calidad del software implantado si aporta interesantes beneficios para alcanzar los objetivos de costes, plazos y calidad final

del producto final. Por ello, a lo largo de este proyecto se recogen las principales características del proceso de pruebas a través de algunos de los más importantes modelos de madurez y mejora, estructuras organizativas y competencias y perfiles profesionales orientados al desarrollo de las pruebas software con el fin de que, a posteriori, se facilite la selección de una combinación óptima de éstas para el desarrollo de un proceso de V&V en una determinada organización.

1.2 Objetivos

El propósito de este proyecto fin de carrera es el de realizar un análisis de los procesos de verificación y validación software con el objeto de determinar el estado actual de su uso, y elaborar una guía de buenas prácticas que permita seleccionar las mejores opciones para realizar las pruebas de software en función de las características de una organización.

Para lo cual se presentan una serie de objetivos que serán tratados a lo largo de la documentación:

- **Objetivo 1:** Analizar los modelos de referencia a través de la representación de algunos de los más importantes en la actualidad. Con el fin de describir los procesos que influyen en el desarrollo de las pruebas software y su posible adaptación a las distintas organizaciones.
- **Objetivo 2:** Analizar las diferentes formas de organizar las pruebas reflejando las estructuras organizativas más comunes en su implantación.
- **Objetivo 3:** Analizar los perfiles y competencias que definen el marco de las pruebas software como una actividad profesional, y establecer una propuesta propia que permita solventar las posibles carencias que presenten dichos perfiles en la actualidad.

1.3 Fases del desarrollo

Dada la tipología del proyecto, éste se ha centrado en labores de investigación y documentación más que en la experimentación. Por ello, desde un principio se han distinguido tres fases principales que han marcado la línea de trabajo a seguir durante la elaboración del mismo:

- **Búsqueda de información:** A lo largo de esta fase se ha tratado de localizar la información que, a posteriori, servirá para el desarrollo del proyecto.
- **Análisis de información:** Una vez concluida la fase de búsqueda de la información, se da inicio al análisis de la misma con el fin de extraer los

conceptos que más se ajusten al tema en cuestión y así favorecer la comprensión del mismo y el desarrollo de las conclusiones.

- **Obtención de resultados:** Finalmente, una vez asentados los conceptos que enmarcan la verificación y validación se establece una guía útil para su implantación en una organización.

1.4 Medios empleados

Para poder llevar a cabo este proyecto no ha sido necesario la utilización de un gran número de recursos ni materiales. Dado que el mismo se basa en la selección de información y en el análisis que se realiza de ésta, para poder llevarlo a cabo se ha hecho uso de bibliotecas tanto físicas como electrónicas. Lo cual se verá posteriormente desarrollado en la sección 3.1 Fase de búsqueda de información.

1.5 Estructura de la memoria

La estructura que presenta la memoria del proyecto fin de carrera se encuentra dividida en cinco capítulos distribuidos de la siguiente manera:

- **Capítulo 1 - Introducción.** Se define el problema en base a la problemática actual dentro del contexto, y se recoge la motivación del análisis y sus objetivos. Por último, se presenta la estructura seguida en el desarrollo de la memoria.
- **Capítulo 2 - Estado del arte.** Se presenta el estado de la cuestión realizando una visión retrospectiva de la actividad y estableciendo una relación con conceptos afines.
- **Capítulo 3 - Propuesta.** Se presenta la documentación extraída con el fin de establecer el uso de la verificación y validación en la actualidad, en la que se definen una serie de modelos, organizaciones, técnicas y perfiles sobre los cuales se establecerán a posteriori las conclusiones finales.
- **Capítulo 4 – Conclusión y futuras líneas.** Se muestran las conclusiones obtenidas tras realizar el análisis del presente proyecto y las posibles líneas futuras en relación a la propuesta.
- **Capítulo 5 – Bibliografía.** En este último capítulo se presentan las fuentes de información utilizadas durante el desarrollo del proyecto.

Capítulo 2

Estado del arte

2.1 Antecedentes

La industria del software, a diferencia de otras industrias, tiene muy poco tiempo de vida. Pero sin embargo, ha conseguido llamar la atención del mercado debido a dos factores esenciales: su velocidad de crecimiento y su alcance.

Desde sus inicios, existieron personas en distintos campos que vieron el avance que para ellos supondría hacer uso de un software especializado, permitiéndoles así la automatización de sus procesos o la aceleración de los mismos. El impacto que finalmente produjo su utilización, no solo tecnológico sino también social, aceleró la demanda y la investigación de ésta tecnología, lo que ha provocado un gran desarrollo de sus niveles de consistencia y rentabilidad.

Para poder obtener dichos niveles, las actividades de *calidad del software* han ido tomando progresivamente una mayor importancia en el ciclo de vida de los proyectos. De hecho, desde hace ya más de 50 años, en 1930 Walter Shewhart comenzó a trabajar en la mejora de procesos estableciendo los principios del control estadístico de la calidad, ya que entendía la calidad como un problema de variación, el cual podría ser controlado y prevenido mediante la eliminación a tiempo de las causas que lo provocan [SHEW30].

Posteriormente, Glenford J. Myers [MYERS79] en 1979 promovió que la Ingeniería del Software separase las disciplinas fundamentales del desarrollo del software de la verificación y validación del mismo. Concepto que, a la postre, se vio fuertemente

reforzado gracias al desarrollo de su principal técnica, las pruebas. De tal forma que acabaría constituyendo el núcleo del aseguramiento de la calidad [IEEE98].

Años más tarde, los principios de Shewhart fueron refinados por W. Edwards Deming [DEMING86] dando lugar al ciclo Deming, el cual fue utilizado entre otras cosas para la mejora continua de la calidad dentro de una empresa. El ciclo consiste en una secuencia lógica de cuatro pasos repetidos que se deben de llevar a cabo consecutivamente en un proceso. Estos pasos son: Planear, Hacer, Verificar y Actuar.

En 1988, el Dr. Dave Gelperin y el Dr. William C. Hetzel [HETZEL88] realizaron una clasificación basada en los más influyentes modelos de pruebas publicados hasta la fecha. Estableciendo cada periodo, a excepción del primero, en función de hitos específicos en forma de investigaciones. A pesar de esta clasificación, los modelos se presentan de forma evolutiva, acumulando así conceptos conforme se desarrollan. De esta forma se establece [HETZEL88]:

Etapas 1- Hasta 1956 – La orientación de las pruebas a la solución de errores.

Durante los comienzos de la TI, a pesar de ser notoria la aparición de problemas relacionados con el software desarrollado, éstos fueron dados de lado en pro de los aspectos hardware. De hecho, estos últimos resultaron ser objeto de los primeros artículos relacionados con las pruebas dentro del campo las TI.

Por aquel entonces, la programación era vista como una actividad que debía ser desarrollada y probada por el mismo programador. Reflejándose de esta forma el poco desarrollo que presentaban las pruebas del software, pero además existían muchos otros conceptos que no se encontraban claramente diferenciados.

El primer artículo basado en las pruebas del software fue escrito por Alan Turing en 1949 [TURING50]. En este artículo se discutía el uso de afirmaciones que hoy en día se podrían denominar como “*pruebas de corrección*”. Se introdujeron conceptos orientados a comprobar si un programa demostraba inteligencia. Conceptos extraídos del planteamiento por el cual se establecía que un programa era la realización de los requisitos planteados del mismo, y la necesidad de saber si el resultado final cumplía con estos. Turing definió el desarrollo de una prueba con el fin de demostrar el comportamiento inteligente de una aplicación. Para llevar a cabo dicha prueba, Turing estableció que el funcionamiento de una aplicación y de un sistema de referencia, en este caso el ser humano, deberían resultar indistinguibles para un tercero, el cual asumiría el papel del probador.

Etapas 2- 1957-1978 - La orientación de las pruebas a la demostración.

En 1957, Charles Baker distinguió el concepto de “*depuración*” del de “*prueba*” en su revisión “*Mathematical Tables and Other Aids to Computation*” [BAKER57] del libro “*Digital Computer Programming*” de Dan MacCracken. En la cual se destacaba que la verificación de los programas perseguía dos objetivos fundamentales:

- “Asegurar el funcionamiento de la aplicación”

- “Asegurar que la aplicación resolvía el problema para el que había sido planteada”.

Para ello, estableció el concepto de depuración en referencia al primer objetivo y el de prueba al segundo. Además, la confianza que se refleja a través de la expresión del término “asegurar”, resultó frecuentemente utilizado en la definición de los objetivos de las pruebas con el fin de demostrar que el software cumplía con sus requerimientos.

Durante este periodo, se produjo un notable incremento de los desarrollos software en términos de cantidad, coste y complejidad. Lo que condujo a que las pruebas vieses aumentado su peso dentro de la industria debido a la necesidad de asumir grandes riesgos económicos, un mayor desarrollo y un incremento en el mantenimiento.

Por aquel entonces, tanto los dirigentes como los propios usuarios empezaron a tomar consciencia de que la cantidad de deficiencias que presentaban los sistemas informáticos era excesiva, y que su solución y la recuperación de los problemas que ocasionaban resultaban demasiados costosos. Por lo cual, se comenzó a dar un mayor énfasis a la “*mejora de las pruebas*”, concepto que perseguía dar una mayor efectividad a las pruebas a la hora de detectar problemas antes de la entrega de los productos.

Durante estos años, los conceptos de depuración y de prueba marcaron el esfuerzo por detectar, localizar, identificar y corregir los fallos. Las diferencias entre estas dos actividades se basaban en la definición del éxito (la resolución de los problemas). Pero la aparición del *modelo de detección* unió dichas tareas de modo que todo fallo detectado era incluido en las “pruebas” y todo fallo localizado, identificado y corregido resultaba “depurado”. Aunque la depuración y la prueba reajustaron sus significados, aún era patente la necesidad de hacer más seguras las ejecuciones de los programas.

En esta etapa, debido a que llevar a cabo las pruebas suponía un esfuerzo comprendido de entre un 30% [SPR] y un 50% [BEIZER90], como mínimo, del coste total de un desarrollo software. Autores como Ince [INCE87] y Jessop [RAMA76] llegaron a la conclusión de que si el proceso de pruebas pudiese ser automático se reduciría significativamente su coste.

Por ello, debido a que el peso de la generación de los datos utilizados en la construcción de los casos de prueba del producto software se consideraba crucial para el éxito de la prueba, en base a la imposibilidad de obtener un programa software completamente probado (el número de casos de prueba necesarios para probar un programa software es infinito [MYERS79]). Y que un diseño adecuado de los casos de prueba puede detectar un elevado número de fallos. A mediados de los años 70 distintos autores presentaron metodologías para la *generación automática de datos de prueba orientados a la trayectoria* en las publicaciones: “A formal system for testing and debugging programs by symbolic execution” [BOYER75] de Boyer, “A system to generate test data and symbolically execute programs.” [CLARKE76] de Clarke, “Automated test data generation” [KOREL90], de Korel y “On the automated generation of program test data” [RAMA76] de Ramamoorthy. Estableciendo el método orientado a la trayectoria como un proceso de selección de los posibles recorridos que se pueden realizar a la hora de alcanzar un estado, generando datos pruebas para recorrerlos y teniendo en cuenta que estos puedan ser creados de forma automática.

Etapa 3- 1979-1982 - La orientación de las pruebas a la detección.

En 1979, [MYERS79] Myers describió el modelo orientado a la detección de pruebas y definió éstas como “el proceso de ejecutar un programa con la intención de encontrar errores”. En esta definición, se estableció con el objetivo principal de la detección de fallos en la implementación, pero la preocupación de Myers estaba centrada en la selección de objetivos que establecían que un software no tenía fallos. Ya que uno mismo puede llegar seleccionar inconscientemente datos de prueba que tienen una baja probabilidad de ocasionar fallos en el programa. Por lo tanto, si el objetivo es demostrar que un programa falla, los datos de prueba deberían tener una alta probabilidad de detectarlos.

El cambio en la orientación de las pruebas de la demostración a la detección trajo consigo la asociación de las pruebas con otras actividades de detección de fallos. Tanto en el libro de Myers como en otras publicaciones nacidas en este periodo: “*Software Project Verification and Validation*” [DEUST57] de Deutsch y “*Software Testing and Validation Techniques*” de Miller [MILLER81], se analizó el uso de las pruebas en el desarrollo del software y su empleo en las técnicas de revisión. Otros artículos como los de “*Software Project Verification and Validation*” de Deutsch [DEUST] y “*Life-Cycle Software Validation*” de Howden [HOWDEN] contienen las primeras descripciones de las aproximaciones a la detección combinada de fallos.

En 1982, surge una nueva metodología orientada a la automatización de casos de prueba denominada “aleatoria”. Publicada en “Automatic generation of random self-checking test cases” [BIRD82] por Bird y Muñoz, resultando ésta una técnica muy simple que crea datos aleatorios para cualquier caso que se quiera probar, lo cual presenta un bajo índice de cobertura.

Etapa 4- 1983-1987 - La orientación de las pruebas a la evaluación.

En 1983, el “*Institute for Computer Sciences and Technology of the National Bureau of Standards*” publica una línea de investigación [NBS] especialmente orientada a los sistemas de procesamiento de información federal (FIPS). En la cual se describe una metodología que integra el análisis, la revisión y las actividades de prueba con el fin de ofrecer una evaluación del producto a lo largo del ciclo de vida del software. Estableciendo la necesidad de obtener en cada fase del ciclo de vida un conjunto de actividades y de productos asociados. Esta guía de desarrollo recomienda tres conjuntos de técnicas de evaluación; la básica, la comprensiva y la crítica. Dichas técnicas presentan un desarrollo evolutivo, de tal forma que cada una de ellas contiene lo desarrollado por su predecesora.

La filosofía colectiva de la cual se hace gala en la metodología FIPS, es fácilmente reconocible en la siguiente cita:

“Ninguna técnica de verificación y validación puede garantizar la corrección, es decir, un software libre de errores. Sin embargo, una cuidadosa elección de técnicas para un proyecto específico puede ayudar a asegurar el desarrollo y el mantenimiento de la calidad del software de un proyecto”

Pero FIPS no fue el único modelo que surgió en el periodo del 83-87. En este periodo fueron creados dos estándares relativos al desempeño de buenas prácticas a lo largo del desarrollo de las pruebas en la ingeniería del software. El primero de ellos, se encontraba relacionado con la documentación del proceso y el segundo hacía referencia a las pruebas unitarias.

En 1979, dado el desarrollo de las pruebas hasta ese momento, Glenford J. Myers [MYERS79] promovió que la Ingeniería del Software separase las disciplinas fundamentales del desarrollo del software de la verificación y validación, disciplina que abarcaría las pruebas del software. El desarrollo de esta disciplina se basó en las pruebas para dar lugar técnicas como las inspecciones y revisiones; y estrategias como las de caja negra y blanca. A pesar de esto, el concepto de verificación y validación no estaba claro. Ambos términos resultaban habitualmente confundidos por lo que Boehm [GARCIA06] los matizó, estableciendo la verificación a través de la cuestión: “¿Se está construyendo el producto de la manera correcta?” y la validación a través de: “¿Se está construyendo el producto correcto?” con el fin de orientar sus respectivas actividades.

Ese mismo año, un grupo del comité técnico de ingeniería del software del IEEE comenzó a trabajar en un estándar para la documentación de las pruebas del software. Este proyecto no se encontraba orientado a estandarizar las mejores prácticas realizadas en aquel entonces, en él se buscó un consenso en la forma de realizar la documentación de las pruebas. Para lo cual, la documentación fue tomada como un sistema de datos estructurados que debería satisfacer la información y el acceso a los requerimientos de sus usuarios. Como resultado, el estándar ANSI/IEEE STD 829-1983 [IEEE83] fue publicado en 1983 especificando el contenido y el formato de ocho documentos estándares. En los cuales se tuvo especial consideración con los aspectos relativos a la modularidad, cohesión, acoplamiento, usabilidad y facilidad de revisión de las pruebas.

Las principales diferencias existentes entre la propuesta ofrecida en el estándar ANSI/IEEE STD 829-1983 y las prácticas realizadas por aquel entonces, residían en las especificaciones de la planificación y el diseño de las pruebas. Un plan de pruebas según establece este estándar basa sus objetivos en función de la identificación de riesgos, el establecimiento de una estrategia general, la definición de una estructura para sus tareas, la distribución de los recursos y sus responsabilidades, el calendario de desarrollo y los posibles contratiempos que puedan surgir. Además, propone la identificación y la descripción de los casos específicos y los procedimientos de prueba con el fin de diferenciar las especificaciones dadas. Pero muchos de los planes de pruebas realizados hasta entonces incluían tanto tareas de planificación como de diseño. Lo cual retrasaba considerablemente la planificación, limitando de esta forma la elección de estrategias.

A parte de diferenciar la especificación de casos y procedimientos, este estándar propone una especificación para el diseño de pruebas. Estableciendo una analogía entre las pruebas y la especificación de la arquitectura del software. Y centrando su propósito en la organización del conjunto de pruebas, de tal manera que se establece una relación directa de éstas con los requisitos del software. Lo cual abre una ventana a la adaptación de las pruebas al software, concepto que normalmente no se encontraba reflejado.

Un año más tarde, en 1984 el congreso del gobierno americano aprobó la creación de un organismo de investigación para el desarrollo de modelos de mejora para los problemas en el desarrollo de los sistemas de software, y evaluar la capacidad de

respuesta y fiabilidad de las compañías que suministran software al Departamento de Defensa (DoD), dando lugar al Instituto de Ingeniería del Software (SEI). En él se comienza un año más tarde a trabajar en un marco de madurez de procesos que permitiese evaluar a las empresas productoras de software.

Años más tarde, un segundo grupo perteneciente al IEEE comenzó a desarrollar un estándar basado en las pruebas unitarias. Como resultado, el estándar ANSI/IEEE STD 1008-1987 [IEEE88] fue publicado en 1987. En él se especifican las fases, actividades, tareas y documentos que comprenden las pruebas unitarias. Continuando la línea que establecía un fuerte paralelismo entre el desarrollo del software y el de las pruebas. Este estándar enfatiza la necesidad de diseñar un conjunto de pruebas, además establece la necesidad de realizar dos tipos de documentos a la hora de implementar las pruebas unitarias: una especificación del diseño de pruebas y un informe del resumen de las mismas. Estos requisitos fomentan el diseño de las pruebas y representan su principal diferencia con las prácticas hasta el momento más comunes, ya que, normalmente, las pruebas de unitarias pasaban desapercibidas.

Un año después de comenzarse a desarrollar el estándar orientado a las pruebas unitarias, en 1985 sus autores introdujeron aquel proceso a lo largo de los distintos niveles de prueba existentes, dando como resultado una metodología conocida como “*el proceso de evaluación y de pruebas sistemáticas*” (STEP) [HETZEL88]. Esta metodología definió un sistema de tareas de pruebas, productos y roles con el fin de dar consistencia y salvar costes a la hora de alcanzar los objetivos propuestos en las pruebas. STEP se basa en un modelo de prevención del ciclo de vida paralelo al desarrollo software, estableciendo una secuencia de actividades como la planificación, el análisis, el diseño, la implementación, la ejecución y el mantenimiento de las pruebas.

Etapa 4- 1988- - La orientación de las pruebas a la prevención.

En 1989, Watts Humphrey y Ron Radice extendieron los principios desarrollados por Deming para su aplicación a la industria del software a través de sus trabajos en IBM y en el SEI.

Ya en la década de los 90, es publicado el libro “Software Testing Techniques” [BEIZER90] el cual hace gala de un gran catalogo de técnicas de prueba. En él, Beizer comenta que “el acto de diseñar pruebas es una de las técnicas más efectivas a la hora de prevenir de defectos”, extendiendo así la definición de las pruebas como concepto de prevención de errores y promoviendo la realización de pruebas en las primeras fases del desarrollo. En ese momento, dado nivel de formalismo que poseían las pruebas, se detalla la definición del proceso de v&v en el IEEE/ANSI 610.12-1990 [IEEE90] estableciendo la verificación como “el proceso de evaluar un sistema o un componente con el afán de determinar cuando los productos de una fase del desarrollo satisfacen las condiciones impuestas al comienzo de la misma.”; y la validación como “el proceso de evaluar un sistema o un componente durante o al final del proceso de desarrollo con el afán de determinar si satisface los requerimientos especificados”.

En ese mismo año, se introduce otra técnica basada en la automatización de los casos de prueba con la publicación de la metodología orientada al objetivo en “Automated test data generation” de Korel [KOREL90]. La cual se establece como el proceso de generar

datos de prueba con el fin alcanzar un determinado estado sin importar el camino que recorran estos.

En 1991, Hetzel [HETZEL91] estableció el proceso de pruebas como la planificación, el diseño, la implementación y la ejecución de las pruebas y sus entornos. Incorporando todas las ideas que hasta la fecha habían considerado a las pruebas como un proceso gestionado, es decir, un ciclo de vida de vida implicado con las pruebas [BEIZER90].

Pero 1991 fue un año especialmente significativo ya que también vio la luz el proyecto comenzado años más tarde por el SEI, el Modelo de Madurez de las Capacidades para el Software (SW-CMM, Capability Maturity Model for Software). El cual, basado en el trabajo de Humphrey, constituye un marco de referencia de la capacidad de las organizaciones de desarrollo de software en el desempeño de sus diferentes procesos, proporcionando una base para la evaluación de la madurez de las mismas y una guía para implementar una estrategia para la mejora continua de los mismos.

2.2 La calidad del software

A lo largo de la evolución de las pruebas, su aplicación en los desarrollos software resulta esencial a la hora de garantizar la calidad software, ya que estas técnicas representan una revisión final de las especificaciones, del diseño y de la codificación. A medida que los clientes se han vuelto cada vez más selectivos y han comenzado a rechazar los productos poco fiables o que realmente no dan respuesta a sus necesidades, la calidad del software ha ido ganando peso en la ingeniería del software y con ella las pruebas.

Pero, sin una definición clara, concisa y medible de lo que es la calidad del software, no es posible tomar buenas decisiones de negocio respecto al uso de los recursos, ni en qué áreas mejorar la calidad, ni que herramientas y técnicas utilizar para la mejora de la misma. Existen diferentes puntos de vista para definir calidad de software. Desde el punto de vista del cumplimiento de los requerimientos Roger Pressman define la calidad de software como:

“El cumplimiento de los requerimientos funcionales y de rendimiento explícitamente definidos, de los estándares de desarrollo explícitamente documentados y de las características implícitas esperadas del desarrollo de software profesional.” [PRESS05]

Por otra parte, IEEE para la ingeniería de software define la calidad del software como:

“El grado con el cual un sistema, componente o proceso cumple con los requerimientos y con las necesidades y expectativas del usuario [WALACE89].”

Como primera aproximación a la calidad del software es importante diferenciar entre la calidad del producto y la calidad del proceso de desarrollo. No obstante, las metas que

se establezcan para la calidad del producto van a determinar las metas a establecer para la calidad del proceso, ya que la calidad del producto se va a encontrar íntimamente ligada a la calidad del proceso de desarrollo. Sin un buen proceso de desarrollo resulta prácticamente imposible obtener un buen producto [WALACE89].

Con el fin de establecer un conjunto de propiedades sobre los productos y los procesos del software que permitan establecer su nivel de calidad surgen los modelos de calidad. Es importante señalar que no basta con tener en cuenta la calidad una vez se ha finalizado el producto. Ésta debe ser considerada en todos sus estados de evolución (especificaciones, diseño, código,...). Además si surge cualquier problema durante su desarrollo que perjudique su calidad, cuanto antes sea atajado el coste ocasionado será menor. Por lo cual, es necesario establecer una serie de actividades con el fin evaluar la calidad de los productos y los propios procesos desarrollados que establecen el control de la calidad. El *proceso de verificación y validación* que es tratado a lo largo de este proyecto cobra una especial importancia dentro de estas actividades.

Algunas personas piensan que la calidad puede lograrse definiendo estándares y procedimientos organizacionales de calidad que comprueben si estos estándares son seguidos por el equipo de desarrollo. Su argumento es que los estándares deben encapsular las buenas prácticas, las cuales conducen inevitablemente a productos de alta calidad. Los gerentes de calidad fomentan a los equipos a responsabilizarse en la calidad de su trabajo y desarrollar nuevas formas para mejorar la calidad. Resultando estos estándares y procedimientos como las bases para la gestión de la calidad.

La gestión de la calidad es particularmente importante para los equipos que desarrollan sistemas grandes y complejos. La documentación de la calidad es un registro de que es hecho por cada subgrupo en el proyecto. Ésta se estructura en tres actividades principales:

- **Garantía de calidad.** El establecimiento de un marco de trabajo de procedimientos y estándares organizacionales que conduce a software de alta calidad.
- **Planificación de la calidad.** La selección de procedimientos y estándares adecuados a partir de este marco de trabajo y la adaptación de éstos para un proyecto software específico.
- **Control de calidad.** La definición y fomento de los procesos que garanticen que los procedimientos y estándares para la calidad del proyecto son seguidos por el equipo de desarrollo de software.

A lo largo de las diferentes actividades que se desarrollan para la consecución de un software de calidad, el proceso de pruebas toma especial importancia. Ya que gracias a este se obtendrán los resultados necesarios para poder tomar las decisiones oportunas que aseguren la calidad del proyecto.

2.3 El proceso software

A lo largo de este capítulo recurrentemente ha sido tratado el concepto de proceso de verificación y validación. Para poder abordar con totales garantías de qué se trata y cuál es su comportamiento dentro de una organización es importante, en primer lugar, conocer su naturaleza. Por ello, conocer la implicación que tiene en una organización el establecimiento de un proceso software facilita la comprensión de cada uno de sus tipos específicos, y este en este caso la verificación y validación lo es.

Un proceso software complementa el concepto de ciclo de vida en el sentido de que éste último define el esqueleto y la filosofía para llevar a cabo un proceso software, pero no es suficiente para guiar y controlar un proyecto de desarrollo y/o mantenimiento. En él se establece un conjunto coherente de políticas, estructuras organizacionales, tecnologías, procedimientos y artefactos que son necesarios para concebir, desarrollar, instalar y mantener un producto software [FUGG00].

La necesidad de participación humana de forma creativa y la ausencia de acciones repetitivas hacen que ni el desarrollo ni el mantenimiento del software sean procesos de fabricación, pero existen algunas similitudes entre ambos tipos de procesos que resultan útiles para comprender los procesos software con una perspectiva más amplia. Al igual que los procesos de fabricación, los procesos software constan de dos sub-procesos interrelacionados: el proceso de producción y el proceso de gestión [LEOD90].

El proceso de producción se relaciona con la producción y el mantenimiento del producto propiamente dicho, mientras que el proceso de gestión proporciona los recursos necesarios para el proceso de producción y lo controla. Esto último es posible si el proceso de producción devuelve información al proceso de gestión sobre su comportamiento.

Además, la gestión tiene que cumplir con los estándares actuales que existen en el entorno; es decir, el entorno exterior influye también indirectamente en el proceso de producción. Finalmente, los procesos de producción y de gestión explotan tecnologías que también vienen del entorno.

A la hora de desarrollar un proceso software se debe de tener en cuenta una serie de aspectos esenciales [CANF04]:

- Tecnologías de desarrollo y de mantenimiento del software, que aportan las herramientas e infraestructuras necesarias para hacer posible crear y mantener productos software complejos que satisfagan las necesidades actuales y futuras.
- Métodos y técnicas para el desarrollo y el mantenimiento software, que suponen el soporte metodológico esencial para aprovechar de manera eficiente las tecnologías y realizar con éxito las actividades de desarrollo y mantenimiento del software.

- Comportamiento organizacional, es decir, la ciencia de las organizaciones y de las personas es útil, en general, los proyectos software se llevan a cabo por equipos de personas que tienen que ser coordinados y dirigidos dentro de una estructura organizacional eficiente.
- Economía, ya que los proyectos de desarrollo y mantenimiento de software no son esfuerzos autónomos sino que, como pasa con cualquier otro producto, el software debe estar dirigido a satisfacer las necesidades de clientes/usuarios reales.

2.3.1 El ciclo de vida

Los modelos de ciclo de vida del software sirven como definiciones de alto nivel de las fases que ocurren durante un proceso. No tienen como objetivo brindar información detallada, pero sí proveer las principales actividades y sus interdependencias. Ejemplos de modelos de ciclos de vida son el modelo en cascada, el de desarrollo evolutivo, desarrollo iterativo e incremental o modelo espiral [IEEE04].

Los modelos de procesos son importantes porque proporcionan un orden en el cual el proyecto debe realizar sus tareas. Entre los primeros modelos propuestos está el modelo en cascada creado por Royce en 1970. En este modelo las etapas se representan una tras otra en el tiempo: Requerimientos, Diseño, Implementación, Prueba y Mantenimiento. Para comenzar una etapa se debe haber completado la anterior. El mayor problema con este modelo es que no refleja realmente la forma en que se desarrolla el software, ya que el software se desarrolla con cierto grado de repetición que la cascada no incluye [PFLE01].

Actualmente, los modelos más modernos, intentan reflejar este comportamiento dividiendo el desarrollo en fases, realizando el desarrollo en forma iterativa e incremental [PFLE01]:

- En los modelos de procesos en fases, el sistema se diseña de modo que puede ser entregado en piezas. Esto permite que los usuarios dispongan de ciertas funcionalidades mientras las otras están siendo desarrolladas.
- En el desarrollo incremental, el sistema, tal como está especificado en el documento de requerimientos, es dividido en subsistemas de acuerdo con su funcionalidad. Se comienza con un subsistema pequeño y se agrega funcionalidad en cada nueva versión.
- En el desarrollo iterativo se entrega un sistema completo desde el principio y luego se mejora la funcionalidad de cada subsistema con cada versión.

A fin de poder establecer a posteriori las diferentes etapas que se puedan dar en el proceso de verificación y validación, a modo de ejemplo, a continuación se proponen dos de los más comunes modelos del ciclo de vida [SOMM05]:

Modelo en Cascada (Bennington)

El modelo en cascada de Bennington está basado en el ciclo convencional de una ingeniería, el paradigma del ciclo de vida abarca las siguientes actividades:

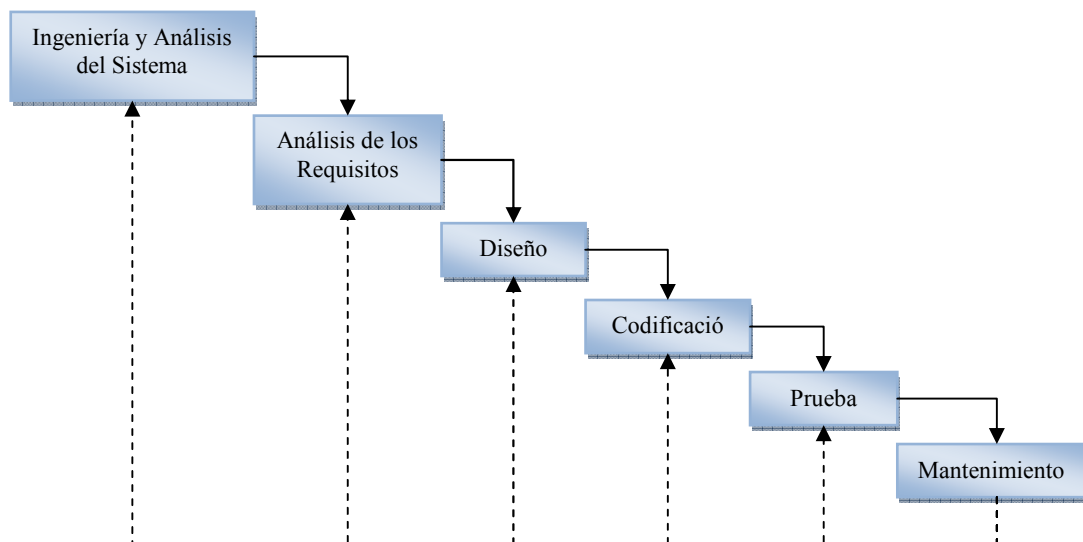


Figura 1. Modelo en cascada del ciclo de vida.

Ingeniería y Análisis del Sistema: Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

Análisis de los requisitos del software: el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

Diseño: el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

Codificación: el diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

Prueba: una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

Mantenimiento: el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que se hayan encontrado errores, a que el software deba

adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

Modelo V

El modelo V tiende a ser muy relacionado con el modelo de cascada puesto que es una evolución del mismo. Donde en la Figura 2 es fácil observar como en su primera mitad es similar al Modelo en Cascada, y la otra mitad tiene como finalidad hacer pruebas e integración asociado a cada una de las etapas de la mitad anterior. Estableciéndose los planes de prueba como el nexo entre el desarrollo y la verificación

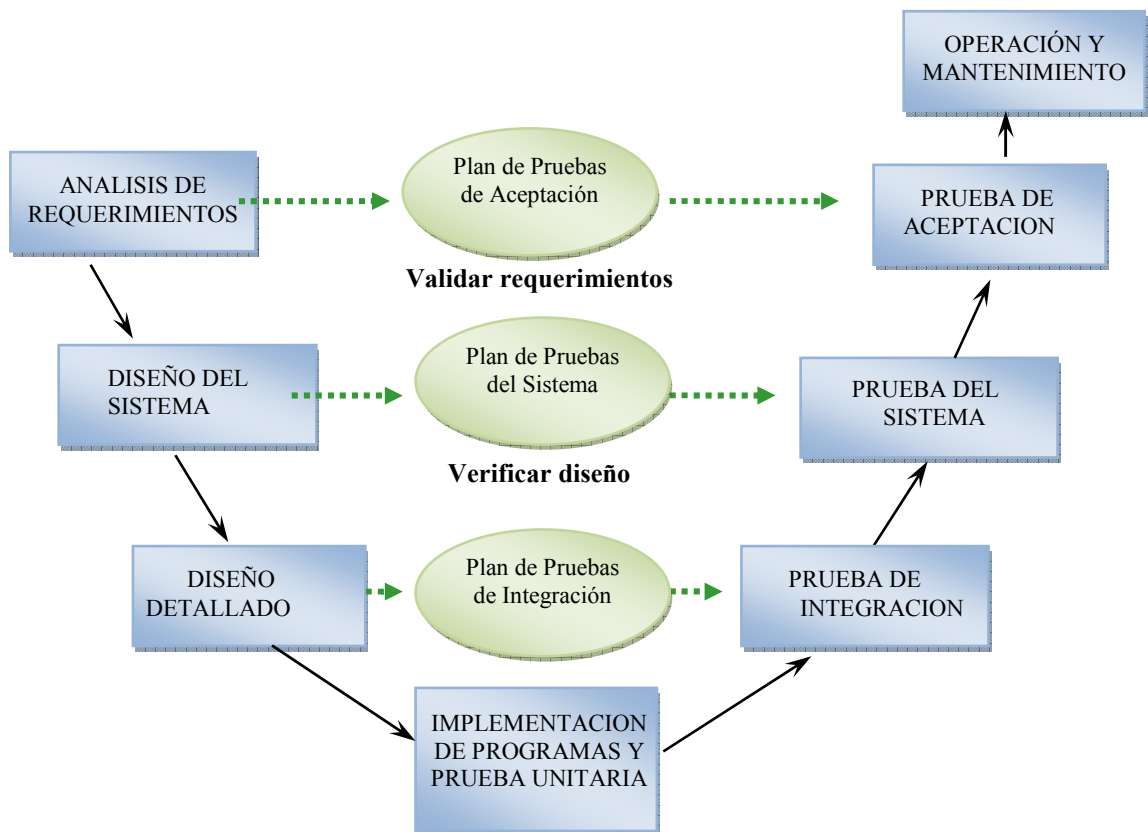


Figura 2. Modelo-V del ciclo de vida.

A través de estos modelos es posible apreciar cómo el proceso de verificación y validación tiene un peso significativo en el desarrollo software y permiten conocer en qué momento se realizarán sus actividades. Aunque en todos no es igual, si es notable un aumento en su presencia en los modelos más actuales. De hecho, existen modelos de proceso como el conocido como el de verificación de la entrada y salida de las actividades ETVX, orientados a representar cómo cualquier fase del desarrollo software debe desarrollarse siguiendo los criterios de v&v, con el fin de minimizar y prevenir el retraso entre la creación de errores y la detección de los defectos.

2.4 El proceso de verificación y validación

El proceso de desarrollo adoptado por un proyecto dependerá de los objetivos y metas que tenga el mismo. Para poder alcanzar dichos objetivos se han desarrollado distintos modelos de ciclo de vida, pero en cualquiera de ellos será necesario un proceso que asegure la calidad del producto durante su desarrollo. Para ello, al final de cada fase del ciclo de vida debería comprobarse que el trabajo realizado hasta ese momento cumple con los objetivos previstos. Este es el punto clave, en el que tiene lugar la evaluación del producto, donde se decide si está o no preparado para pasar a la siguiente fase. De esta forma, si hay errores y son detectados, será más eficiente corregirlos que si se descubriesen en etapas más avanzadas.

El objetivo final del proceso de verificación y validación es comprobar que el sistema está hecho para un propósito, al cual intenta llegar aplicando técnicas específicas conocidas como: las pruebas y las revisiones

El *proceso de validación y verificación (V&V)* [SOMM05] es un conjunto de procedimientos, actividades, técnicas y herramientas que se utilizan paralelamente al desarrollo de software, con el fin asegurar que un producto resuelve el problema inicialmente planteado. Más específicamente [SOMM05] [PRESS05]:

La verificación

La verificación comprueba la consistencia del software con respecto a especificaciones y requisitos, es decir, si responde a la pregunta: ¿se ha construido correctamente el software?

- El proceso determina si los productos resultantes de una fase del ciclo de vida software cumplen los requisitos establecidos en la fase anterior.
- El proceso determina si el producto resultante es completo, consistente y correcto para comenzar la siguiente fase.

La validación

La validación comprueba si lo que se ha especificado e implementado es lo que el usuario realmente desea, es decir, si responde a la pregunta: ¿se ha construido el software correcto?

- El proceso determina si el software cumple su especificación.
- El proceso asegura que el software fabricado se comporta como se espera y de acuerdo a las expectativas del cliente.

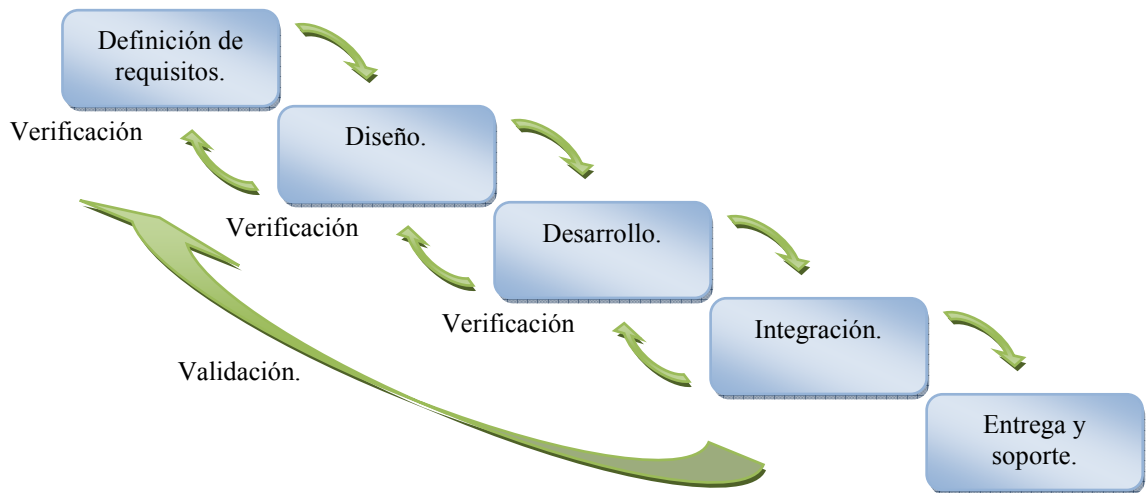


Figura 3. V&V en el ciclo de vida software.

Las tareas de V&V no solo se aplican a productos software, sino que también a los resultantes del proceso de desarrollo, como al análisis y a la especificación de requisitos, por ejemplo. De esta forma, se comprueba que el proyecto es viable y que las especificaciones documentadas son completas, correctas, precisas, legibles, evaluables, y que, en general, responden a las expectativas del cliente [SOMM05]. La V&V del diseño debe garantizar que los requisitos no se encuentran incompletos o incorrectamente diseñados. En el caso de la implementación y la codificación, la V&V de software es comúnmente conocida como las pruebas de software.

Existen muchas definiciones incorrectas sobre las pruebas del software que conducen a una inadecuada aplicación de este proceso, por ejemplo [PRESS05]: “Las pruebas demuestran que no hay errores”, o “Las pruebas demuestran que un programa funciona correctamente”. Según Edgar Dijkstra “Las pruebas pueden demostrar la presencia de errores, no su ausencia”. Por lo tanto, se realizan con el fin de detectar errores que, una vez corregidos, mejoran la calidad o la fiabilidad del mismo. Existen distintos tipos de pruebas en función de la unidad de software a la que se aplique y del objetivo que se persiga, por ejemplo, las pruebas de unidad, de integración, de sistema y de aceptación.

Finalmente, las actividades de V&V son también necesarias durante la operación y el mantenimiento del software. Cuando se realiza un cambio en el mismo, se debe examinar el impacto de éste sobre el sistema y considerar qué actividades se necesitan repetir para garantizar, al menos, la misma calidad en el software antes del cambio.

Otro factor a tener en cuenta es la independencia del proceso de V&V, ya que es una característica reconocida como altamente positiva en distintas áreas, reportando dentro del desarrollo software una serie de características importantes [SOMM05]:

- La separación de intereses, por la cual se evitan conflictos de intereses internos en una organización. Una organización independiente garantiza que requisitos importantes no sean ignorados en el proceso de toma de decisiones.
- La diferenciación de puntos de vista, esto supone una segunda opinión, que siempre es interesante para eliminar ambigüedades u omisiones.

- La efectividad y la productividad, las actividades de V&V son llevadas a cabo por personal experto, con competencias específicas en el área de V&V.

A pesar de estas características, es importante tener en cuenta que la independencia no debe ser un muro infranqueable entre el equipo de desarrollo y el equipo de V&V, por ello es importante fomentar un entorno de participación y de aprendizaje hacia un objetivo común.

2.4.1 Técnicas de verificación y validación

Hoy en día existen dos paradigmas principales que permiten la aplicación de las actividades de V&V en un proyecto, el análisis estático y el dinámico [SOMM05].

El análisis estático necesita de la ayuda de una herramienta que analice la representación estática del sistema (diagramas de flujo, casos de uso, código fuente, otros) con el fin de descubrir errores. Este tipo de análisis implica la no ejecución de modelos, ni del código fuente para determinar las posibles fallas. Se trata por lo tanto de verificar los “productos” obtenidos durante el desarrollo del software. Para llevarlos a cabo es habitual el uso de técnicas de inspección, éstas no resultan normalmente un reemplazo de las pruebas de software, sino que las complementan, enfocándose en aquellas diferencias que no son posibles detectar usando únicamente pruebas.

Por el contrario, el análisis dinámico implica normalmente la ejecución del código fuente del software con datos de prueba, con el fin de evaluar en tiempo real los diferentes escenarios soportados por el programa.

Las técnicas de inspección comprenden el análisis automático del código fuente y la verificación formal. Por el contrario, las técnicas estáticas sólo pueden comprobar la correspondencia entre un programa y su especificación (verificación), no pudiendo demostrar que el software es operacionalmente útil [PRESS05].

2.4.1.1 Técnicas estáticas

Las técnicas estáticas [SOMM05] son las primeras comprobaciones que se aplican al software. Son técnicas cuyo objetivo es el de mejorar la calidad de los productos del software, ayudando a los ingenieros a reconocer y arreglar sus propios errores en etapas tempranas del proceso de desarrollo sin ejecutar código. Para ello, buscan defectos sin ejecutar el código, llevándose acabo una vez se encuentra escrito el código.

Estas técnicas tienen que ver con el análisis y el control de las representaciones del sistema, es decir con los diferentes modelos construidos durante el proceso de desarrollo de software, tales como los documentos de requerimientos, los diagramas de análisis y de diseño y el código fuente. En esta categoría entran, por ejemplo, las revisiones, las inspecciones de programas, la verificación formal (basada en el uso de notaciones formales con base matemática) y las herramientas de análisis estático (muchas de ellas provistas hoy en día como parte del compilador, como por ejemplo la detección de variables no utilizadas, código inalcanzable, etc.).

La **revisión del software** es un proceso o una reunión durante la cual un producto de software es examinado por personal del proyecto, los encargados, los usuarios, los clientes, los representantes del usuario u otros interesados [IEEE97].

En este contexto, el término producto de software engloba cualquier documento técnico, final o parcial, elaborado como un entregable de una actividad del desarrollo del software, pudiendo incluir documentos tales como: los contratos, los planes, los presupuestos, los documentos de los requisitos, las especificaciones, los diseños, el código de fuente, la documentación del usuario, la documentación de la ayuda y de mantenimiento, las especificaciones de prueba y los estándares.

Las revisiones del software se pueden dividir en tres categorías [PRESS05]:

- Las *revisiones de par del software*, las cuales son conducidos por el autor del producto, o por uno o más compañeros del mismo, con el fin de evaluar el contenido técnico y/o la calidad del trabajo.
- Las *revisiones de la gerencia del software*, las cuales son conducidas por los representantes de la gerencia con el fin de evaluar el estado del trabajo realizado y de tomar decisiones con respecto a las siguientes actividades.
- Las *revisiones de la intervención del software*, las cuales son conducidos por el personal externo al proyecto del software, para evaluar conformidad con especificaciones, estándares, acuerdos contractuales, u otros criterios.

Por otro lado, las revisiones atienden a la formalidad que identifica el grado por el cual una actividad cumple con los estándares. El estándar de IEEE. 1028-1997 define el acercamiento más formal que es seguido por los *walkthroughs*, las revisiones técnicas, las inspecciones y las auditorías del software [IEEE97]. Dicho estándar define un sistema común de actividades que estructura las revisiones formales.

Las revisiones informales

A la hora de llevar a cabo las revisiones, éstas no siempre se encuentran enmarcadas en una estructura para ser llevadas a cabo. Este es el caso de las revisiones informales, entre las que se distinguen [SOMM05]:

- La **revisión del código**, que consiste en una forma de evaluación sistemática del código, normalmente elaborada por el propio desarrollador. Se realiza de forma individual estableciendo una depuración de un determinado código.
- La **programación en parejas**, que consiste en una revisión del código en el que dos personas trabajan juntas, en colaboración, para revisar de forma periódica un determinado producto.

Las revisiones formales

Las revisiones formales siguen una secuencia de actividades basada en gran medida en la inspección del software de Michael Fagan [FAGAN76], las cuales son detalladas a continuación:

1. Planificación: La planificación se produce cuando el desarrollador completa su producto, entonces se forma un grupo de inspección y se designa un moderador. El moderador debe asegurarse que el producto satisfaga el criterio de inspección. Por último, se le asignan diferentes roles a las personas que integran el grupo de inspección, así como la planificación de tiempos y recursos necesarios.

2. Familiarización: El paso de familiarización sólo se realiza si los inspectores no conocen el desarrollo del proyecto. Es opcional pero no por ello menos importante, ya que en esta etapa se dará al grupo de inspección el contexto a cubrir por las inspecciones.

3. Preparación: Los inspectores se preparan individualmente para la evaluación en la reunión, estudiando los productos y el material relacionado. En este paso es aconsejable la utilización de listas de chequeos para ayudar a encontrar defectos comunes. El tiempo que pueda llevar esta etapa va a depender de la familiarización del inspector con el trabajo que debe analizar.

4. Examen: En esta etapa, los inspectores se reúnen para analizar su trabajo individual de forma conjunta. El moderador deberá asegurarse que todos los inspectores se encuentran suficientemente preparados. La persona designada como lector presenta el producto interpretando o parafraseando el texto, mientras que cada participante observa en busca de defectos. Es recomendable que este examen no dure más de 2 horas ya que la atención en busca de defectos va disminuyendo con el tiempo. Al terminar con la reunión, el grupo determina si el producto es aceptado o se debe volver a realizar para una posterior inspección.

5. Rehacer: El autor corrige todos los defectos encontrados por los inspectores.

6. Seguimiento: El moderador chequea las correcciones del autor. Si el moderador está satisfecho, la inspección está formalmente completa, y el producto es puesto bajo el control de configuración.

A continuación se presentan una serie de revisiones que cumplen con la secuencia de actividades propuesta por Fagan, las cuales difieren unas de otras principalmente en el grado de aplicación de cada fase [IEEE97]:

- Las *inspecciones* evalúan el código de sentencia en sentencia, centrándose principalmente en los documentos, los modelos y el código. Pueden ser aplicadas en todas las fases, esto es muy importante ya que permite iniciar las V&V antes de escribir una sola línea de código. Las inspecciones se pueden entender como un repaso detallado y formal del trabajo en proceso a través de pequeños grupos de trabajadores, los cuales estudian el producto de forma independiente, y posteriormente se reúnen con el fin de examinar el trabajo en detalle. Los productos no serán considerados óptimos hasta que estén completas la inspección y las correcciones necesarias.

- Los **walkthroughs** son inspecciones, pero en ellas el análisis es presentado por el propio desarrollador y el mecanismo para conducir la inspección es el desarrollo de un escenario sobre el código.
- La **revisión técnica** del software es una forma de revisión de par, en la cual un equipo cualificado examina la conveniencia del producto software para su uso previsto e identifica discrepancias con las especificaciones y los estándares.
- Las **auditorías** son un tipo de revisión del software la cuál es llevada a cabo por uno o más interventores, los cuales no pertenecen a los miembros de la organización. Se trata de un examen independiente de un producto, de un proceso, o de un sistema software con el fin de determinar su conformidad con las especificaciones, los estándares, los acuerdos contractuales u otros criterios.

2.4.1.2 Técnicas dinámicas

Las técnicas dinámicas, también conocidas como **pruebas** [SOMM05], se basan en ejercitar una implementación. Por lo tanto, sólo pueden ser aplicadas si existe una versión operativa o ejecutable del producto. A pesar de que las técnicas de verificación estáticas son usadas cada vez más, la prueba de los programas sigue siendo la técnica predominante en la validación y verificación.

La prueba involucra ejecutar el programa proveyéndole entradas con el fin de detectar la mayor cantidad de defectos posibles. Son diseñadas con el propósito de revelar la presencia de defectos, por ello se dice que un caso de prueba es exitoso cuando los logra poner en evidencia. Además, son consideradas como la única técnica de validación para los requerimientos no funcionales, debido a que el software debe de ser ejecutado para ver su comportamiento, proporcionando confianza en el software.

Con el fin de obtener pruebas lo más sistemáticas posibles, se busca identificar un conjunto representativo de comportamientos del programa, determinados por subclases del dominio de las entradas, los escenarios y los estados, diferenciando dos enfoques [PRESS05]:

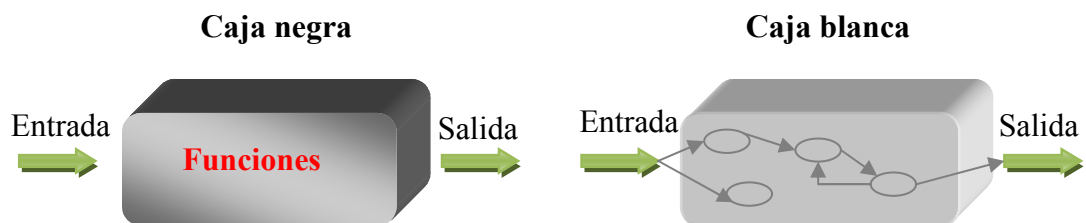


Figura 4. Técnicas de caja negra y blanca.

Caja Negra (Funcional): Las técnicas de caja negra se distinguen porque los casos de prueba se basan sólo en el comportamiento de entrada/salida, diferenciándose:

- Las *fragmentaciones equivalentes*, donde el dominio de las entradas se divide en subconjuntos equivalentes respecto de una relación especificada.
- El *análisis de los valores límite*, similar a la anterior pero los valores de entrada de los casos de prueba se eligen en las cercanías de los límites de los dominios de entrada de las variables.

Caja Blanca (Estructural): Las técnicas de caja blanca se encuentran basadas en la información de cómo el software ha sido diseñado, entre ellos:

- *Los criterios de cobertura:*
 - La cobertura de sentencias, por la cual cada sentencia se ejecuta al menos una vez.
 - La cobertura de decisiones, por la cual cada decisión tenga, por lo menos una vez, un resultado verdadero y, al menos una vez, uno falso.
 - La cobertura de condiciones, por la cual cada condición de cada decisión adopte el valor verdadero al menos una vez y el falso al menos una vez.
 - El criterio de decisión/condición, por el cual se cumplan a la vez el criterio de condiciones y el de decisiones.
 - El criterio de condición múltiple, por la cual la evaluación de las condiciones de cada decisión no se realiza de forma simultánea.
- Los *grafos de flujo* utilizados para la representación del flujo de control. La estructura de control sirve de base para obtener los casos de prueba. El diseño de casos de prueba debe de estar basado en la elección de caminos importantes que ofrezcan una seguridad aceptable de que se descubren defectos.

Las pruebas del sistema

Las pruebas del sistema implican la integración de dos o más componentes, los cuales implementan funciones del sistema o características y posteriormente se prueba dicho sistema integrado [SOMM05]. En un proceso de desarrollo iterativo, las pruebas de sistema se ocupan de probar un incremento que va a ser entregado al cliente, por el contrario en un proceso en cascada, éstas mismas se ocupan del sistema completo. En la mayoría de los sistemas complejos, existen tres fases distintas de pruebas del sistema [SOMM05]:

- Las *pruebas de integración*, en las cuales el equipo de pruebas tiene acceso al código fuente del sistema. Cuando se descubre un problema, el equipo de integración intenta encontrar la fuente del mismo, e identificar los componentes que tienen que ser depurados.
- Las *pruebas de entregas*, en las cuales se prueba una versión del sistema que podría ser entregado a los usuarios. Aquí, el equipo de pruebas se ocupa de validar que el sistema satisface sus requerimientos y de asegurar que es fiable. Las pruebas de entregas son normalmente pruebas de caja negra.

Cuando los clientes se implican en las pruebas de entregas, éstas a menudo se denominan *pruebas de aceptación*. Si la entrega es lo suficientemente buena, el cliente puede entonces aceptarla para su uso.

- Las *pruebas de rendimiento*, las cuales se realizan una vez se ha integrado el sistema completamente, de esta forma es posible probar las propiedades emergentes del sistema, propiedades tales como el rendimiento y la fiabilidad. Este tipo implica estresar al sistema realizando demandas que se encuentran fuera de los límites del diseño del software. Son particularmente relevantes para los sistemas distribuidos basados en una red de procesadores.

Las pruebas de componentes

Las pruebas de componentes o pruebas unitarias son el proceso de prueba de los componentes individuales del sistema. Existen diferentes tipos de elementos que pueden ser objeto de prueba [SOMM05]:

- Los *métodos o funciones individuales* dentro de un objeto son el tipo más simple de componente, y sus pruebas son un conjunto de llamadas a dichas rutinas con diferentes parámetros de entrada. Las pruebas unitarias consisten en encontrar errores basándose en la estructura o en las especificaciones, verificando el funcionamiento individual de partes del software.
- Las *clases de objeto* que tienen varios atributos y métodos, para las cuales se deberían diseñar las pruebas con el fin de proporcionar cobertura para todas las características del objeto, incluyendo pruebas aisladas de todas sus operaciones asociadas, la consulta de sus atributos y la ejecución de sus distintos estados.
- Los *componentes compuestos* formados por diferentes objetos o funciones, los cuales poseen una interfaz definida utilizada para acceder a su funcionalidad. Las pruebas se ocuparán principalmente de demostrar que se comportan conforme a su especificación.

2.5 Las pruebas software

Como se ha podido observar a lo largo del capítulo, el término de pruebas del software ha sufrido una constante evolución a lo largo de la historia de las TI. Un concepto clave para su definición ha sido por el cual éstas son consideradas como un proceso, o una serie de ellos, encaminados a asegurar que el código desarrollado cumple con el propósito con el cual fue diseñado y no tiene resultados indeseados. (Myers 2004). El IEEE por otra parte los estableció como “*el proceso de operar un sistema o un componente bajo unas condiciones específicas, observando o registrando los resultados obtenidos con el fin de realizar a posteriori una evaluación de ciertos aspectos clave.*”; y

como “el proceso de analizar un atributo del software con el fin de detectar la diferencia existente entre las condiciones requeridas y las presentadas”

Analizando las distintas definiciones se puede extraer un concepto común a todas ellas. En él se destaca el proceso de establecer datos de entrada al producto software e inspeccionar los resultados obtenidos. Fundamentalmente, las pruebas del software responden a dos preguntas, ¿se ha obtenido un buen producto?, ¿se ha desarrollado de forma correcta? Lo cual, asegura que las pruebas software cubren el ciclo completo de vida del producto desde su desarrollo hasta su mantenimiento. Este concepto da lugar al proceso de verificación y validación del software.

Con el fin de desarrollar las pruebas de una manera óptima, los casos de prueba cobran especial importancia. Estos son un conjunto de entradas, condiciones de ejecución y resultados esperados que son desarrollados con el fin de cumplir un determinado objetivo. De tal forma que ejecute un camino específico dentro del programa o que verifique su adecuación con los requisitos [IEEE90].

El principal objetivo de las pruebas es el de encontrar errores en el programa, tantos como sea posible. Entorno al concepto de error existen distintos términos que poseen ciertos aspectos en común y a menudo inducen a confusión. Por ello, el estándar IEEE/ANSI 610.12-1990 [IEEE90] establece:

- **Falta:** acción humana que produce un resultado incorrecto.
- **Defecto:** paso, proceso o definición de datos incorrectos en una aplicación a consecuencia de una falta.
- **Fallo:** resultado incorrecto, es el resultado de un defecto.
- **Error:** el valor del resultado incorrecto.

Las pruebas del software se diferencian del resto de actividades del proceso de desarrollo software en que éstas son un proceso destructivo. Lo cual significa que el objetivo del probador, que es quien realiza las pruebas, es el de descubrir errores, es decir, encontrar el mayor número de errores. Por ello, un buen caso de pruebas es el que permite localizar un gran número de errores. El trabajo de probador de software resulta demasiado especializado por lo que todo el mundo no está preparado para llevarlo a cabo, debido a que la naturaleza humana es mayormente constructiva, no destructiva.

Otro factor que resulta especialmente condicionante para su labor es la dificultad de probar el código desarrollado por uno mismo. La persona que ha desarrollado un código establece un punto de partida desde el que considera que todo funciona de forma correcta. Resulta difícil pensar que lo que ha hecho uno mismo está mal. Esta es la razón por la cual los desarrolladores y los probadores normalmente se encuentran distribuidos en grupos distintos y por lo que también no suelen llegar a tener un mutuo entendimiento. Pero que en lugar ello sería necesario comprender que los probadores se encuentran añadiendo un valor al producto, descubriendo errores e informando de ellos tan pronto como sea posible. Su intención no es la de perjudicar la producción.

2.5.1 Los principios de las pruebas

A pesar de que las pruebas del software son una tarea técnica, siempre son llevadas a cabo por alguien, por lo que tanto el aspecto humano como el económico resultan especialmente importantes. Myers [MYERS79] define diez principios fundamentales:

- Una parte necesaria de un caso de prueba es la definición de los resultados esperados.
- Un programador debe evitar probar su propio desarrollo.
- La organización que desarrolla no debería probar sus propios programas.
- La inspección detalla de cada resultado de las pruebas.
- Los casos de prueba deben de ser escritos teniendo en cuenta las condiciones inválidas o inesperadas, de la misma forma que se hacen con condiciones validas y esperadas.
- La comprobación de que un programa hace lo que debería, y que no hace lo que no.
- Evitar los casos de prueba de un solo uso a no ser que el programa también lo sea.
- No realizar una planificación de pruebas asumiendo que no se encontraran errores.
- La probabilidad de encontrar más errores en una sección es proporcional al número de errores encontrados hasta el momento.
- Las pruebas son tareas que implican con una gran carga creativa e intelectual.

2.6 La mejora de los procesos software

Tal y como se ha visto a lo largo del capítulo, existe un estrecho vinculo entre la calidad del proceso de desarrollo y la calidad de los productos desarrollados por el mismo. En consecuencia, muchas compañías de ingeniería del software han tomado el camino de la mejora de los procesos del software para mejorar su producto. La mejora de procesos significa entender los procesos existentes y cambiarlos para mejorar la calidad del producto y reducir costes y tiempo de desarrollo. Para ello, centran sus esfuerzos en la optimización de los procesos para mejorar la calidad del producto y, en particular, en reducir el número de defectos en el software entregado [SOMM05].

Esta situación, sumada a que la verificación y la validación juegan un papel importante en el aseguramiento de la calidad de un proceso o un producto. Y debido a

que ésta por sí misma es considerada un proceso software, hace que la verificación y validación sea objetivo de la mejora de procesos software.

La mejora de procesos está basada en la suposición de que la calidad del proceso desarrollado es crítica para la calidad del producto. Estas nociones de mejora de procesos provienen del ingeniero estadounidense W.E.Deming [DEMING86], quien trabajó en la industria japonesa después de la Segunda Guerra Mundial para mejorar la calidad. Deming, entre otros, introdujo la idea del control estadístico de la calidad. Ésta se basa en medir el número de defectos en los productos y relacionar estos defectos con el proceso. Éste se mejora con el propósito de reducir el número de defectos en el producto hasta que sea repetible, es decir, hasta que los resultados del proceso sean predecibles y el número de defectos se reduzcan. Después se estandariza e inicia un ciclo de mejoras de calidad. Por ello, unas de las ideas básicas en la mejora de procesos se trata de un ciclo de acciones correctivas y preventivas llamado "ciclo de Deming" o modelo PDCA, referido a los siguientes cuatro pasos:

- **Planear (plan):** definir los objetivos a alcanzar y planificar cómo implementar las acciones.
- **Hacer (do):** implementar las acciones correctivas.
- **Controlar (check):** verificar que se logre el conjunto de objetivos.
- **Actuar (act):** según los resultados obtenidos en el paso anterior, tomar medidas preventivas.

A lo largo de la historia, han sido varias las instituciones, como el caso del SEI, que teniendo como motivación la mejora de la calidad a través de la mejora del proceso. Entre los resultados de estas entidades se encuentran recapitulaciones y catálogos de buenas prácticas y modelos que han dado lugar a los *modelos de mejora de los procesos software*.

2.6.1 Modelos de mejora de los procesos software

Orientados a la mejora de procesos, los *modelos SPI (Software process improvement)* dotan de un marco de referencia para desarrollar software acorde con la planificación establecida, mientras simultáneamente mejora la capacidad del desarrollador de crear mejores productos. Un modelo de procesos software puede ser usado por una organización para asegurar su madurez e identificar y dar prioridad a las áreas con mayor importancia. En este contexto, el término madurez está referido a la capacidad de una organización de llegar a un estado, definido, continuo y optimizado.

Existen numerosos modelos SPI en la actualidad, uno de los más extendidos en la industria del software es CMMI, el cuál será presentado con posterioridad [AMESC02]. Por otra parte, los modelos de mejora suelen guardar cierta estructura común respecto a la realización de actividades de mejora. Siguiendo la estructura desarrollada por el grupo SOGETI en la cual se muestran las distintas actividades de un proceso de mejora, se establece [DUTTA99]:

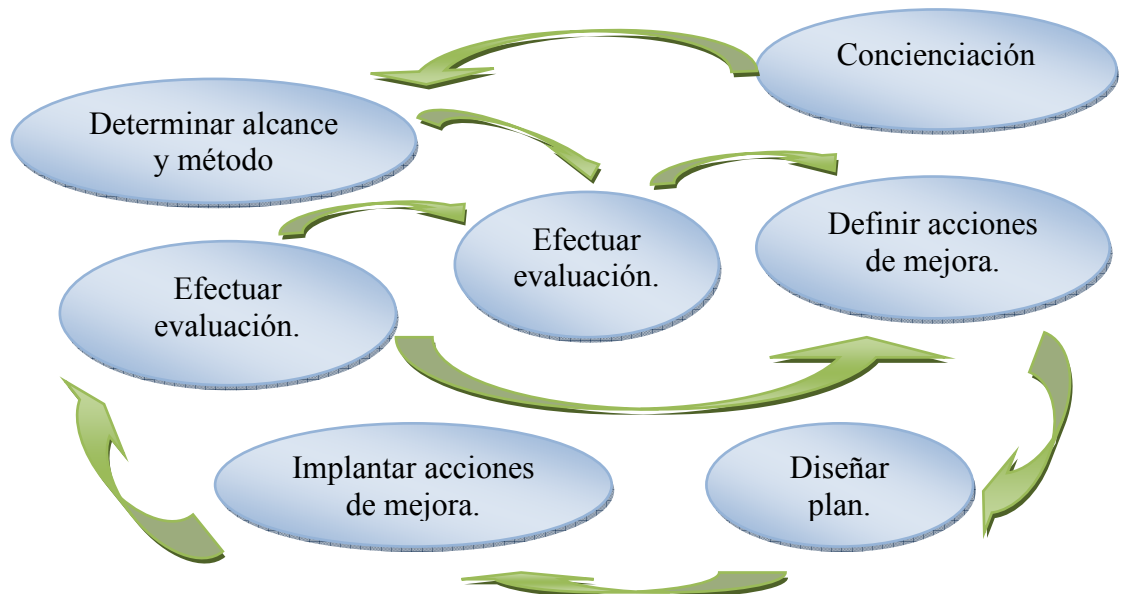


Figura 5. Actividades de un proceso de mejora.

Conciliación. La primera actividad de un proceso de mejora consiste en tomar conciencia de la necesidad de mejorar un proceso software. Una de las principales razones para realizar dicha mejora se trata de la existencia de problemas en los procesos que deben de ser solucionados aunque también destaca la mejorara de la eficiencia de un determinado proyecto. La concienciación implica que las diferentes partes implicadas en el proceso acuerden mutuamente las líneas generales y se comprometan al proceso de cambio. Este compromiso no solo debe realizarse al comienzo del proceso, sino que debe permanecer a lo largo de todo el proyecto, realizando un esfuerzo continuo.

Determinar el alcance y método. Una vez que se ha tomado conciencia de la necesidad de mejorar el proceso, es necesario tratar de determinar cuáles son los objetivos y el alcance de la mejora, así es posible establecer qué pruebas se van realizar, los procesos que son susceptibles de mejora, el tiempo, el esfuerzo y los costes necesarios para hacer frente al proceso de mejora.

Efectuar la evaluación. El desarrollo de esta actividad conlleva evaluar la situación actual del proyecto obteniendo, como resultado de la evaluación, los puntos fuertes o débiles de los procesos.

Definir las acciones de mejora. Después de recoger la información obtenida en las actividades anteriores en forma de objetivos de mejora se procede a la elaboración de acciones que posibiliten la mejora paulatina de los mismos.

Diseñar el plan. Una vez establecidas las acciones que se van a realizar en el paso anterior, se diseña un plan de implantación de las acciones de mejora a corto plazo. En este plan se determina en qué momento deben realizarse las acciones de con el fin de alcanzar los objetivos marcados. El plan se dirige tanto a las actividades relacionadas con el contenido de las mejoras del proceso como a las actividades generales necesarias para realizar el proceso de cambio en la dirección adecuada.

Implantar acciones de mejora. Una vez discutido el plan, se pone en marcha, es decir se lleva a cabo la implantación de las acciones de mejora. Debido a que durante esta

actividad, las consecuencias del proceso de cambio tienen un mayor impacto, hay que prestar mucha atención a la comunicación. Sin duda, puede haber cierta resistencia al cambio, por lo que deberá afrontarse y discutirse abiertamente. Se debe medir hasta qué punto se han ejecutado las acciones y si han resultado un éxito.

Efectuar Evaluación. Finalmente, después de implantar las acciones de mejora se realiza una evaluación que permite comprobar en qué medida las acciones fueron implantadas con éxito, así como evaluar en qué medida se alcanzaron los objetivos iniciales. A partir de estas observaciones se adoptará la decisión sobre la continuación del proceso de cambio.

2.6.2 Modelos de mejora de los procesos software

La mejora de los procesos puede y debe aplicarse a la totalidad de los mismos para obtener un beneficio en la calidad del proyecto. En particular, atendiendo a la calidad del software, los procesos de pruebas cobran especial importancia. Debido a esta razón, y a raíz de los intentos realizados con el fin de obtener una mejora de la calidad de los productos software a través de los modelos de desarrollo, se observó la existencia de una grave carencia en el tratamiento ofrecido a los procesos orientados a las pruebas. A consecuencia de esta carencia, las pruebas se convertían en un proceso complicado e incluso incontrolable. Pudiendo llegar suponer un proceso demasiado duradero y dotado de un alto número de recursos. Además, el hecho de poder superar los costes previstos y de aportar una percepción insuficiente de la calidad del sistema probado, suponía un riesgo para el propio proceso de negocio. Todo esto, impulsó la creación de nuevos modelos de mejora de los procesos orientados a las pruebas, con el fin de aumentar la calidad del producto ofertado y de su desarrollo. Debido a que los modelos orientados a la mejora de los procesos de prueba provienen, en gran parte, de la necesidad de perfeccionar los modelos de mejora de los procesos de desarrollo del software ya existentes, poseen múltiples similitudes y conceptos compartidos, pero aplicados a las pruebas del software [WALACE89]. Pero estos últimos basan su desarrollo en la consecución de procesos de pruebas maduros.

Para poder desarrollar este concepto en primer lugar es necesario tener claro el significado de madurez. En el trabajo “The Capability Maturity Model” [PAULK95] publicado por Paulk y Weber se ofrece una buena aproximación a dicho concepto. En él se comparan y se contrastan las características que presentan los comportamientos considerados maduros frente a los inmaduros de las organizaciones de software [PAULK95]. También se describen un conjunto de conceptos fundamentales en la madurez de los procesos software. Para completar este concepto Paul define un proceso de pruebas maduro como un proceso gestionado, medido, seguido y efectivo. De esta forma, que el proceso de pruebas se considere como un proceso gestionado es interpretado en términos de planificación, gestión de recursos, dirección, control y de componentes organizativos [SUWAN96]. Un proceso de pruebas maduro se establecerá como un proceso básico de la organización y, apoyado por la dirección, formará parte de la cultura organizacional. Por último, la madurez de los procesos de prueba debe ser fácilmente comprendida y debe tener la capacidad de continuo crecimiento y mejora. Para poder aplicar el concepto de mejora de forma específica a las pruebas se establecen una

serie de atributos, que su consecución se atribuyen a altos niveles de madurez por parte de un proceso de pruebas [SUWAN96]:

Un conjunto de políticas definidas. La existencia de un conjunto de políticas de pruebas bien definidas y documentadas que se aplican en toda la organización. La cuales deben de ser apoyadas por la alta dirección con el fin de conseguir su institucionalización y su integración en la cultura organizativa.

Un ciclo de vida de las pruebas. El proceso de pruebas esta en gran parte basado en actividades orientas a la ejecución de las mismas, por lo tanto es necesaria la existencia de un ciclo de vida bien definido, que establezca un conjunto de fases y actividades integradas dentro del ciclo de vida del desarrollo. El ciclo de vida de las pruebas abarca todas las actividades de éstas, incluyendo la planificación de las pruebas, las revisiones de sus productos y la documentación relacionada. El ciclo se debe aplicar a todos los proyectos.

Un proceso de planificación de las pruebas. La existencia de un proceso de planificación de las pruebas bien definido y documentado utilizado en toda la organización permite la especificación de los objetivos y de las metas de las pruebas. Además de la asignación de recursos, el diseño de las pruebas, el desarrollo de los casos de prueba, así como la documentación de las tareas de pruebas, sus costos, y la programación de pruebas.

Un grupo de pruebas. La existe de un grupo de pruebas establece la posición de un probador de forma que ésta se encuentra definida y apoyada por la alta dirección. Introduciendo además sus competencias y la formación específica que adquieren, ofreciendo una trayectoria profesional a modo de motivación.

Un grupo de mejora del proceso. La existencia de un grupo dedicado a la mejora del proceso de pruebas, los cuales pueden formar parte de un grupo SQA o ser algún componente del grupo de pruebas, fomenta que el proceso de pruebas se encuentre bien definido y medido. Así mismo, el grupo de mejora de las pruebas puede ejercer el liderazgo oportuno a la hora de concretar los procesos y aplicar de esta forma una mejora incremental de las técnicas empleadas y evaluar su impacto.

Un conjunto de métricas de prueba. La organización debe tener un programa de medición a través de un conjunto de métricas que permitan definir, almacenar y analizar. Estas métricas son usadas para dar soporte a las acciones de mejora apropiadas para cada circunstancia.

Herramientas y equipo. El uso de herramientas facilita las labores del grupo de pruebas de recopilación y análisis de los datos. El grupo de mejora del proceso de prueba evalúa las potenciales herramientas, y supervisa las tecnológicas relacionadas con la integración de las herramientas en el entorno de la organización.

Un control y seguimiento. El proceso de pruebas debe ser seguido y controlado por los gerentes de pruebas con el fin de supervisar su progreso y poder emprender acciones correctivas cuando ocurran problemas. Además, de esta forma es posible evaluar el rendimiento y la capacidad del proceso, utilizándose para ello técnicas cuantitativas que determinan la capacidad del proceso y su efectividad.

Control de la calidad del producto. Durante el desarrollo del proceso se deben utilizar métodos estadísticos con el fin de establecer la calidad del producto frente a los estándares. La calidad del producto se controla, los defectos son rastreados, y el análisis de causalidad se aplica para la prevención de defectos.

2.7 Estructura organizacional

Históricamente, el compromiso de las organizaciones en la elaboración de desarrollos en los cuales se favorezca la calidad del producto final, se ha visto limitado a compañías centradas en desarrollos críticos. El resto, han basado sus pruebas en los propios desarrolladores o en pequeñas organizaciones de estos últimos [OVD].

Con la evolución de la industria del software y la creciente competencia en la mayoría de los segmentos, la calidad del producto ha ido ganado peso como reclamo de clientes. Por esta razón, hoy en día muchas empresas realizan un esfuerzo significativo para adecuar organizaciones destinadas a obtener mayores niveles de calidad. Lo cual ha supuesto a su vez, una evolución en la forma de entender la organización de las pruebas y su estructura [OVD].

El proceso de pruebas resulta una actividad compleja, la cual involucra la interacción de personas, procesos y herramientas con el fin de desarrollar un producto. En la práctica, al igual que el desarrollo de software, es llevado a cabo por equipos que consisten en un conjunto de individuos.

Muchos de los estudios relacionados con la obtención de procesos y resultados de calidad en el desarrollo software se centran en la relación del personal, los procesos y su ámbito, pero a menudo ignoran uno de los factores más influyentes en el desarrollo del software, el personal y su estructura organizativa, como en el caso que se va a tratar a lo largo de este proyecto, la v&v [BROOKS95].

De hecho la Ley de Conway enunciada en 1968, establece que “Las organizaciones que diseñan sistemas están limitadas a producir diseños que son copias de las estructuras de comunicación de estas organizaciones.” [BROOKS95] Es decir, que los equipos que logren una buena comunicación en el mundo real tendrán más probabilidades de crear interfaces de software de mejor calidad entre sus subsistemas. Por otro lado, los equipos que tengan problemas de comunicación interpersonal construirán interfaces entre los subsistemas que reflejarán en estas dificultades.

La Ley de Conway es una interesante observación sobre el comportamiento humano, sus relaciones interpersonales y sus consecuencias en el trabajo en equipo y la creación de software. La consecuencia es ya conocida: los buenos equipos generan buen software. Es por esto que la problemática real del desarrollo de software es la creación de grandes equipos humanos, que colaboren y se comuniquen entre sí para luego poder crear el mejor software posible. Las organizaciones rígidas que no están dispuestas a reorganizarse para crear diseños óptimos (usualmente por miedo a crear una organización peor) terminan produciendo diseños pobres que reflejan la organización pre-existente.

De forma similar, Fred Brooks introdujo en “Mythical Man Month” que la calidad del producto se ve fuertemente afectada por la estructura organizativa implementada. Además con el empuje que presenta el desarrollo global donde existen equipos distribuidos por todo el mundo el impacto de una estructura organizativa eficiente se hace más que notable

Estos conceptos llevados al ámbito exclusivo de las pruebas software llevan a establecer la necesidad de que los equipos de pruebas deban de estar debidamente estructurados, definiendo las funciones y responsabilidades que permitan a los probadores centrarse en sus funciones, evitando la superposición de tareas y evitando la incertidumbre de qué miembro del equipo debe realizar las funciones [PEAR06]. Una forma de dividir estos recursos es mediante la especialización en áreas de aplicación y áreas funcionales.

Las estructuras organizacionales se encuentran directamente relacionadas con las características del personal que las componen y llegan a cobrar una especial importancia desde el punto de vista de la efectividad, debido a que un diseño apropiado de dichas estructuras puede proporcionar resultados óptimos a través de la utilización de distintos equipos de trabajo que de forma coordinada establezcan un mejor control del trabajo. Incluso estos pueden dar una visión de carrera profesional a los miembros de los equipos.

Para establecer diferentes estructuras normalmente se presta atención a dos conceptos fundamentales, el tipo de organización y su distribución geográfica. En función de la primera es posible establecer dos tipos de organizaciones [PEAR06]:

- **Organizaciones de producto:** Las cuales desarrollan productos software y se desarrollan en fases de diseño, desarrollo, pruebas y mantenimiento. Las pruebas forman parte de estas fases y los grupos de pruebas suelen pertenecer a la propia organización.
- **Organizaciones de servicio:** Éstas no poseen una responsabilidad completa sobre el producto final. En el contexto de las pruebas, se tratan de organizaciones externas que proveen de servicios a otras organizaciones que los requieren. Además, prestan personal especializado y actualizado para la realización de pruebas.

El segundo factor que representa un importante rol en la decisión del tipo de estructura organizativa a implementar es la distribución geográfica de los equipos. Tanto las organizaciones de producto como las de servicio pueden desarrollarse de forma distribuida o en un único emplazamiento [PEAR06]. En un equipo localizado, todos los miembros se encuentran situados en el mismo lugar, mientras que un equipo distribuido estos pueden encontrarse en distintas localizaciones. Éstos últimos introducen diferentes factores como los culturales que afectan la estructura de la organización.

Existen innumerables formas de implementarlas estas organizaciones, pero deben de ser estudiadas y seleccionadas en función de las características de la empresa, acomodándose a las necesidades comerciales de la misma. Así, las organizaciones de pruebas tienen que estar estructuradas en concordancia con la política, la cultura corporativa, la cultura de calidad, las características y el nivel de conocimiento de los

participantes y los riesgos de los productos [MELLON02]. A lo largo de este proyecto se presentan cuatro organizaciones con gran peso en la actualidad: la metodología convencional, los grupos SQA, los grupos de pruebas y el outsourcing.

2.8 Competencias y perfiles profesionales

Las capacidades de las que hace gala un equipo de prueba pueden afectar al éxito o al fracaso del proceso. Por ello, para estos equipos, no es suficiente con ser técnicamente competentes, demostrando dotes en el uso de técnicas y herramientas de prueba. Dependiendo de la complejidad del contexto de las pruebas, el equipo deberá incluir miembros que posean un conocimiento detallado del mismo. Este conocimiento les permitirá crear casos de prueba efectivos y datos efectivos para la realización de las mismas.

Es importante señalar que las actividades de pruebas resultan a menudo complejas, es por este motivo por el cual precisan que el personal dedicado a su elaboración posea una amplia diversidad de conocimientos, con el fin de comprender el alcance y la profundidad necesaria para elaborar una estrategia de pruebas eficiente. Por ello, la delimitación de las capacidades desarrolladas por estos profesionales resulta en la actualidad de vital importancia, para lo cual se busca un conjunto de competencias y perfiles específicos que permitan focalizar la formación de los futuros profesionales.

Los conceptos de competencias y perfiles profesionales guardan una estrecha relación, tanto es así que autores como Martínez Mut [MUT05] y Levi-Leboyer [LEVO00] establecen diferencias entre ambas destacando las competencias como *“la integración de conocimientos profesionales, habilidades y actitudes en la realización eficaz de las tareas propias de un puesto de trabajo en un contexto concreto”*. Mientras que un perfil profesional como *“la identidad profesional de las personas que, con una titulación académica, llevan a cabo una determinada labor y explica las funciones principales que dicha profesión cumple”*.

Para que todos los miembros de un equipo puedan estar al tanto de las tareas que se deben realizar y saber quién va a ser el responsable de cada una de ellas, es necesario definir y documentar las funciones y las responsabilidades de los miembros del equipo de pruebas. Estos, deben de ser comunicados, tanto de forma verbal como escrita a todos los miembros del equipo. Por ello, los perfiles comentados anteriormente pueden ser concretados en forma de distintos *roles* dentro de un proyecto, especializándose así en una serie de competencias mucho más específicas que facilita su adaptación dentro de un proyecto (Figura 6). La identificación de las funciones asignadas a los miembros de un equipo de prueba permite entender con claridad que individuo es el responsable de cada área del proyecto. Y en particular, permite a los nuevos miembros del equipo saber con quién contactar en caso de que surjan problemas.

En el desarrollo de los perfiles profesionales y de sus competencias es común realizar una distinción entre los tipos éstas, estableciéndose como específicas y genéricas. Las primeras se desarrollaran a lo largo de distintos perfiles, mientras que las segundas definirán o concretarán la funcionalidad del perfil o rol en el que se desarrollen:

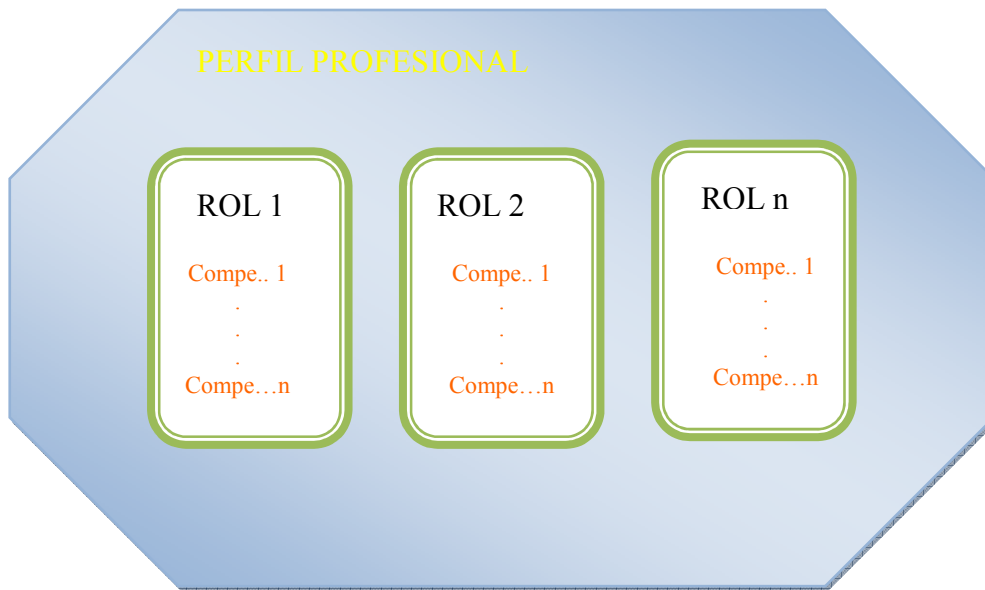


Figura 6. Esquema de la estructura de un perfil profesional.

A pesar de lo necesario que resulta una buena especificación de las capacidades de los profesionales dedicados a las pruebas software, hoy en día muchos de los ingenieros informáticos que salen por primera vez al mercado laboral, a pesar de presentar grandes habilidades de desarrollo, demuestran graves carencias en lo referente a las pruebas y a la depuración del software. Este hecho, es debido en parte al actual énfasis que demuestran los académicos por las labores de desarrollo dejando de lado la prueba de software como una disciplina.

A pesar de los esfuerzos por formalizar la profesión, no solo en lo referido a las pruebas del software sino que a todo el conjunto de la ingeniería del software, aún quedan aspectos que resultan necesarios mejorar. De hecho Career Space, que es un consorcio formado por grandes compañías de TIC (como Cisco Systems, IBM Europe, Intel, Microsoft Europe, Nokia, Nortel Networks, Philips Semiconductors, Siemens AG, Telefónica S.A. y Thales) inste a las universidades europeas a que implanten tanto los nuevos currículos de TIC como el acuerdo de Bolonia, con el fin de ayudar a reducir el vacío existente en Europa respecto a las cualificaciones sobre dichas materias [ANECA].

En un informe denominado “Perfiles de capacidades profesionales genéricas de TIC. Capacidades profesionales futuras para el mundo del mañana” [CARRER], Career Space propone dieciocho perfiles genéricos de puestos de trabajo organizado en las siguientes áreas:

Telecomunicaciones:

- Ingeniería de radiofrecuencia.
- Diseño digital.
- Ingeniería de comunicación de datos.
- Diseño de aplicaciones para el procesamiento digital de señales.
- Diseño de redes de comunicación.

- Software y servicios
- Desarrollo de software y aplicaciones.
- Arquitectura y diseño de software.
- Diseño multimedia.
- Consultoría de empresas de TI.
- Asistencia técnica.

Productos y sistemas:

- Diseño del producto.
- Ingeniería de integración y pruebas e implantación y pruebas.
- Especialista en sistemas.

Intersectoriales:

- Dirección de marketing de TIC.
- Dirección de proyectos de TIC.
- Desarrollo de investigación y tecnología.
- Dirección de TIC.
- Dirección de ventas de TIC.

Pero, como se puede ver a simple vista, el peso que presentan las labores de calidad y más en concreto, las pruebas parecen pasar casi desapercibidas.

2.9 Resumen del estado del arte

A lo largo de este capítulo se ha venido contextualizando el tema sobre el que va a girar este proyecto fin de carrera, los procesos de verificación y validación software. Para lo cual se han presentado conceptos de directa implicación y se ha introducido el mismo.

Los procesos de verificación y de validación son unas actividades adaptadas al ciclo de vida del desarrollo del software centradas en comprobar que un producto se adapte a sus requisitos. Este es un concepto directamente relacionado con la calidad del software, pero ésta presenta un abanico muy amplio de factores que cumplir. Por ello, una buena definición del concepto de calidad para un proyecto va a facilitar las decisiones de negocio necesarias para alcanzarla, desde los recursos asociados hasta qué técnicas y dónde aplicarlas.

A la hora de implantar un proceso de validación y verificación, hay que tener en cuenta que éste como su propio nombre indica es un proceso. Como todo proceso se compone de una serie de actividades que establecen su desarrollo como la planificación, el diseño, la implantación y su comprobación. Además, comprende una serie de necesidades organizacionales para distribuir y formar al personal que se hará cargo de dichas actividades. Pero para el desarrollo de su actividad este personal deberá hacer gala de una serie de tecnologías y técnicas específicas. En este caso, las técnicas de verificación y validación son las pruebas y las revisiones. Ambas técnicas se encuentran basadas en el concepto de “pruebas”, aunque dentro de la v&v las primeras estén

destinadas a comprobar qué se está desarrollando correctamente y las segundas que lo que se ha desarrollado cumple con sus requisitos.

El concepto de pruebas ha sufrido una gran evolución desde su aparición ganando importancia a medida que han aumentado las necesidades de obtener un software de mayor calidad en la sociedad. Esta evolución ha llevado a las pruebas desde un concepto en las que se las suponía una actividad a realizar por el propio desarrollador y que se encontraban destinadas a demostrar que el producto cumplía los requisitos pactados. A buscar los posibles fallos y a planificar y a diseñar los modelos en busca de una optimización del proceso, que se considera más eficiente si es desarrollado por un agente independiente. Este concepto de optimización que se ha desarrollado en las pruebas ha supuesto desde la creación de metodologías orientadas a hacer más eficiente su actividad como es el caso de la automatización de los casos de pruebas, a la creación de modelos de mejora.

La necesidad de mejorar es un concepto muy amplio y en principio estuvo enfocado a la calidad. Para lo cual, se creó el modelo CMM que permitía medir la madurez de los procesos implantados en una organización. A modo de guía de buenas prácticas, este sirvió de referencia para posteriores modelos no solo centrados en procesos software sino que también aparecerían modelos orientados a las pruebas.

A la hora de desarrollar un proceso de pruebas, no solo se debe prestar atención al propio proceso y a sus productos. Para poder llevarlo a cabo resulta de vital importancia el factor humano. El personal que se dedica a este tipo de labores ha de estar organizado, de forma que el conjunto de las labores realizadas por cada uno resulten productivas. Por ello, es necesaria una buena elección del tipo de organización estructural que se va a llevar a cabo a la hora de implementar dicho proceso. Una elección óptima de la organización favorece la comunicación del personal y una delimitación de tareas que permite llevarlas a cabo de una manera más profesional, ya que se fomenta la especialización. Debido a la necesidad de especialización que implica la formación de grupos de trabajos orientados a labores de prueba, se fomenta la necesidad de establecer una serie de competencias y perfiles profesionales que regulen la profesión. Permitiendo asignar un puesto de trabajo en relación a las habilidades y conocimientos que presente un individuo. Éstos no solo deberán poseer conocimientos comunes al resto de las ingenierías sino que además deberán poseer una serie de conocimiento y habilidades en técnicas, herramientas y metodologías de prueba en función del perfil profesional quieran pertenecer.

Capítulo 3

Propuesta

3.1 Fase de búsqueda de información

A la hora de realizar el proyecto, aunque se han sucedido distintas fases hasta llegar a su conclusión, uno de los puntos clave en su desarrollo ha sido la búsqueda de información que, a posteriori, ha servido como referencia a la hora de analizar la situación de los procesos de verificación y validación software, y de la creación de una propuesta para la elección de una configuración del mismo en una organización.

Dado que el proyecto ha sido planteado en torno a tres ejes fundamentales: los modelos del proceso, su organización y las competencias y perfiles profesionales. Las búsquedas se han realizado en torno a estos tres conceptos. Pero tan importante como saber qué hay que buscar, ha sido saber dónde hay que buscar. La información sobre la que se ha investigado ha girado en torno a tres formatos: los artículos, los libros y las páginas Web. Es por ello, que en primer lugar se estableció un entorno de búsqueda dónde éstas se iban a tener lugar. Según el tipo de documento que se pretendía localizar se seleccionaron las distintas bibliotecas y buscadores:

Artículos.

ACM Digital Library. Ésta es una vasta colección de documentos a texto completo y referencias bibliográficas a través de la cual se accede a las publicaciones periódicas y actas de congresos de la “*association for computing machinery (ACM)*”. En ella se pueden encontrar documentos desde 1950 hasta la actualidad, documentos como: Artículos de revistas, ponencias y actas de congresos, noticias breves, informes.

ScienceDirect – Sciverse (Elsevier). Éste es un metabuscador que integra tres importantes portales: Scopus, Science Direct y Scirus.

Scirus es el metabuscador científico más amplio que ofrece *Elsevier*, rastrea en sitios del tipo .edu .org .ac.uk, .gov y algunos .com. Además, rastrea todo tipo de archivos, repositorios o bibliotecas digitales de libre acceso como Arxiv.org, Pubmed central, RePEc, y además otros portales científicos de acceso de pago como Lexis-Nexis o Science Direct.

Scopus es la mayor base de datos bibliográfica de revistas científicas “*peer review*” y de seguimiento de citas, con 44 millones de documentos. Además, desde Scopus se analizan un alto número de revistas científicas en lengua española, 1,200 Open Access journals y no sólo revistas: se analizan libros, tesis, conferencias y congresos.

Science Direct recoge 2.500 títulos de revistas de alto nivel científico, producidas por Elsevier, con acceso directo al texto completo. Contiene igualmente 11.000 títulos de e-books. Se pueden hacer todo tipo de búsquedas.

IEEE Xplore. Ésta es la plataforma electrónica de las publicaciones científicas de la organización sin ánimo de lucro, IEEE. La Biblioteca Electrónica en línea de IEEE/IEE acerca las publicaciones de la IEEE y la IEE al usuario. En ella se encuentran referenciados artículos, documentos de conferencias, y estándares técnicos desde 1988 a la fecha. Contiene más de 12,000 publicaciones con más de 2 millones de imágenes de páginas de 12 años hacia atrás.

Wiley Online Library. Ésta colección fue creada a partir de la fusión de las colecciones “*Blackwell Sinergy*” y “*Wiley Interscience*” en junio de 2008, ofrece un acceso cómodo e integrado a más de 4 millones de artículos de 1500 revistas, 9000 libros y cientos de obras de referencia, protocolos de laboratorio y bases de datos de John Wiley & Sons e importantes sellos editoriales como “*Wiley-Blackwell*”, “*Wiley-VCH*” y “*Jossey-Bass*”.

ISI Web of Knowledge (Cross Search). El contenido de ISI Web of Knowledge es multidisciplinario, y soporta las investigaciones dirigidas por organizaciones corporativas, gubernamentales, académicas y sin ánimo de lucro de todo el mundo. Desde un único punto de entrada, “*ISI Web of Knowledge*” integra “*ISI Web of Science*” e “*ISI Current Contents Connect*” con datos de patente (“*Derwent Innovations Index*”), información sobre procedimientos (“*ISI Proceedings*”), literatura sobre las ciencias de la vida (“*BIOSIS Previews*”), química estructural (“*ISI Chemistry*”), herramientas analíticas y de evaluación (“*ISI Essential Science Indicators*”, “*ISI Journal Citation Reports*”), herramientas de gestión bibliográfica (“*EndNote*”, “*ProCite*”, “*Reference Manager*”), y

contenidos web evaluados. ISI contiene información bibliográfica de más de 16.000 revistas, libros y ponencias internacionales en el campo del arte y las humanidades, las ciencias sociales y las ciencias. Una parte importante de estos 16.000 documentos son revistas, 8.600.

Google Scholar. Éste es un buscador de Google especializado en artículos de revistas científicas, enfocado en el mundo académico, y soportado por una base de datos disponible libremente en Internet que almacena un amplio conjunto de trabajos de investigación científica de distintas disciplinas y en distintos formatos de publicación. Fue lanzado al público en versión Beta el 18 de noviembre de 2004. El índice de “*Google Scholar*” incluye las revistas más leídas en el mundo científico con excepción de Elsevier.

Libros.

Safari online books. “*Safari® Tech Books*” es una fuente de información de Tecnologías de la Información. Es la unión de “*O'Reilly Media*” y “*The Pearson Technology Group*”. Ofrece una valiosa colección de referencia del contenido de estos dos publicadores de libros de Tecnología de la Información. Es una fuente de información que provee de un nivel de autoridad, flexibilidad y simultaneidad, cubriendo las tecnologías más esenciales para los usuarios como: certificaciones, computación de empresas, Java, Linus/Unix, desarrollo de tecnologías de Internet, Windows, XML, y muchos más. Con más de 3,700 libros los usuarios pueden buscar simultáneamente entre cientos de libros, ahorrándose tiempo y rápidamente encontrando la información que necesitan.

Google Books. Se trata de un servicio de Google que busca el texto completo de los libros que Google escanea, convierte el texto por medio de reconocimiento óptico de caracteres y los almacena en su base de datos en línea. El servicio era conocido como Google Print cuando fue introducido en la Feria del Libro de Fráncfort en octubre de 2004. El 9 de octubre de 2009, Google anunció que el número de libros digitalizados se encontraba por encima de los 10 millones.

Biblioteca Uc3m. En la biblioteca de la universidad Carlos III es posible encontrar recursos electrónicos, realizar búsquedas en varios recursos al mismo tiempo, consultar su colección de revistas electrónicas, y acceder al texto completo si se dispone de suscripción, con un amplio de catálogo de artículos y revistas en formato físico. Además, a través de la metabúsqueda es posible obtener un listado de los recursos por MATERIA/TIPOLOGÍA y realizar una búsqueda simultánea en los recursos que marques.

Webs.

Google. Se trata del buscador más potente, es una herramienta que permite rastrear e indexar miles de millones de páginas web y documentos asociados, y realizar una clasificación de su relevancia para cualquier término que el usuario busque.

Una vez seleccionados los medios a través de los cuales se realizarían las búsquedas, se procedió a la selección de los términos de búsqueda mediante los cuales se obtendrían los sondeos para seleccionar la información más relevante para cada cuestión. Para ello, teniendo en cuenta que dado que la mayor parte de la documentación escrita se encuentra en inglés, estos términos deberían seguir dicha pauta. Algunos ejemplos de estos términos que han servido para localizar la información manejada a lo largo del proyecto se reflejan en la siguiente tabla:

Modelos	Organizaciones.	Competencias
Test maturity model	Testing methods	Software quality skills.
Test Maturity Model integrated.	Mangemnt of software testing	Competencias y perfiles. Profesionales informática.
Capability maturity model integration	Software testing work teams.	Software engineer competencies.
Test process improvement TMap.	Software verification and validation groups.	Software testing professional profile.
Test Management Approach.	Structural organization of software testing.	Test engineering skills.
Maturity of software testing.	Responsabilities of software testing.	Software testing responsibilities
Software testing improvement.	Software quality assurance team.	SQA skills.
Quality of software testing process.	Software testing outsourcing.	Software testing curriculum.
Effective methods for software testing	Independence software testing groups	Software testing competences.
		Verification and validation skills

Tabla 1. Términos de búsqueda.

Para potenciar su rendimiento, éstos fueron combinados y utilizados en distintos idiomas (inglés y castellano). De esta forma se completa la primera parte del trabajo dedicado a la búsqueda de información.

A medida que el proyecto fue tomando forma, la información que solicitada se volvía mucho más concisa. Para dar respuesta a las nuevas necesidades planteadas, las búsquedas se vieron depuradas a través de la introducción de elementos técnicos en las nuevas búsquedas, por ejemplo: las búsquedas de organizaciones de prueba outsourcing fueron completadas gracias a las búsquedas relacionadas con “*Testing factories*”, que es un grupo específico de éstas.

Además, con el fin de completar la bibliografía y obtener una información de calidad, no solo se ha realizado un especial hincapié en aquella información que se encontraba mejor valorada en términos de relevancia por las distintas bibliotecas y buscadores, sino que se han profundizado las búsquedas a partir de la bibliografía manejada en dichos recursos. Resultando búsquedas directas a partir de un título o de un ISBN en concreto.

Esta metodología se ha seguido en la localización de artículos y libros principalmente, sin embargo la búsqueda de páginas web se ha visto centrada en la búsqueda de organizaciones ya conocidas a través de la información obtenida. Así de esta

forma, sus búsquedas han sido basadas en sus nombres o abreviaturas, por ejemplo SEI o “*Software Engineering Institute*”

Por último, cabe destacar, que a lo largo de todo el proceso de búsqueda de información se ha llevado una labor de clasificación, con el fin de facilitar la posterior adaptación de la información localizada al análisis. Por ello, aunque se ha venido hablando de la estructuración de la información entorno a tres ejes fundamentales, éstos resultaban demasiado extensos, por lo que se optó por su clasificación.

A la hora de realizar búsquedas en torno a los modelos de mejora de las pruebas, se partió desde el concepto de que la información iba a girar sobre los principales modelos utilizados en la actualidad. Apareciendo así cinco puntos clave de apoyo a la hora de dividir la información, estos son: CMMi, TMMi, TMM, TPI y TMaP. Además, se mantuvo otro punto de apoyo centrado en información más generalista, la cual permitiese presentar el conjunto de los modelos.

Para abordar la organización estructural del proceso de pruebas, fueron seleccionados cuatro tipos diferentes de organizaciones sobre las que giraría la información y las cuales serían tomadas como ámbitos de clasificación. Así, se establecieron los grupos de pruebas, los grupos SQA, el outsourcing y la metodología convencional.

Por último, dentro del ámbito de las competencias se produjo una clasificación más reducida de la información, estableciéndose entorno a las competencias por un lado y a los perfiles profesionales por otro.

3.2 Fase de análisis de la información

La segunda fase que se ha llevado a cabo en el desarrollo del proyecto, en concordancia con el objetivo principal del mismo, trata el análisis de los procesos de verificación y validación del software con el objeto de determinar su estado actual.

Para desarrollar dicho análisis y siguiendo las pautas que se viene marcando a lo largo del proyecto en torno a los tres ejes de información propuestos acerca de los procesos de verificación, la información se ha analizado en función de las características más importantes que presentan los modelos, las organizaciones y las competencias en el ámbito de las pruebas. Las cuales se presentan de la siguiente manera:

Modelos de referencia.

La información orientada a los modelos de pruebas tiene el propósito de presentar los principales modelos y metodologías que engloban los procesos de verificación y validación del software. De esta forma se establece CMMI como un modelo base, ya que trata la madurez del conjunto de los procesos software estableciendo pautas de cómo se tienen que establecer los procesos en una organización para que presenten unas condiciones óptimas de calidad. Además, esta información es reforzada a través de modelos como TMM, TMMi y TPI que centran sus esfuerzos en la mejora del propio

proceso de pruebas, presentando pautas específicas para el desarrollo de dicho proceso. Por último, se hace una referencia a la metodología de pruebas TMap, la cual establece un amplio estudio del desarrollo de un proceso de pruebas software. A través de la documentación obtenida en este capítulo se pretende dar una visión de cómo se establecen los procesos de v&v en la actualidad.

Con el fin de destacar a lo largo del análisis factores que influyan en la implantación de un determinado modelo en una organización, se han determinado una serie de características sobre las que se va a centrar la información presentada más adelante. Para ello se han seleccionado:

- **Tipo:** El tipo de un modelo consiste en conocer la clase u objetivo del mismo, de esta forma se van a presentar tres formas distintas:
 - Modelo de calidad y mejora: El cual se orienta a la mejora del conjunto de los procesos software con el fin de aumentar la calidad de desarrollo en una organización.
 - Modelo de mejora de las pruebas. El cual, basado en el concepto de mejora de los anteriores modelos, centra sus esfuerzos en mejorar el desarrollo específico del proceso de pruebas.
 - Metodología de proceso de pruebas. A diferencia de los anteriores, en él se marcan las pautas a seguir para la implantación y desarrollo de un proceso de pruebas.
- **Vinculación modelo/método:** Tanto los modelos como las metodologías pueden estar relacionados unos con otros de forma que complementen conceptos. Esto puede ser un factor clave a la hora de seleccionar nuevos modelos que aplicar en una organización debido a que disminuye el impacto ocasionado.
- **Modelo en segmentos:** El desarrollo de un modelo basado en segmentos asegura que conseguidos unos objetivos, éstos serán comunes a un conjunto de procesos.
- **Modelo continuo:** El desarrollo de un modelo continuo facilita que una organización seleccione los procesos que quiere fomentar y actuar directamente sobre ellos.
- **Detalle de pruebas:** Según los modelos presentados, diferenciar si un modelo se encuentra exclusivamente centrado al proceso de pruebas o no puede marcar los objetivos de la organización y posibilitarle entrar en ciertos mercados.
- **Reconocimiento de organismos internacionales:** Siguiendo las pautas marcadas en el punto anterior, el reconocimiento internacional marcado por instituciones como ISO o ESI puede suponer la apertura de ciertos mercados.
- **Complejidad:** La complejidad que marca un modelo se basa principalmente en la cantidad de tareas y gestiones que implica, de modo que expresa el coste de una organización por implantarlo desde cero.

Análisis de las diferentes formas de hacer pruebas.

La información referente a la estructura organizacional adoptada en el desarrollo de las pruebas presenta distintas formas en las que actualmente se dividen, se agrupan y se coordinan las actividades y recursos orientados a realizar labores de v&v dentro una organización. Para lo cual, se han seleccionado tanto pequeñas estructuras como otras que pueden llegar cobrar cierta envergadura. En el caso de las que presentan un tamaño menor se destacan los grupos de pruebas y las metodologías tradicionales. Los grupos de pruebas son organizaciones que cuentan con labores específicas de pruebas, mientras que la metodología tradicional se basa en el concepto de que el propio desarrollador sea quien las realice. Por otro lado, se presentan estructuras que pueden abarcar una mayor diversidad de funciones, pero que su actividad se desarrolla desde la v&v como es el caso del outsourcing de pruebas y los grupos SQA. El outsourcing, es un concepto basado en la externalización de servicios y a lo largo de este capítulo se presenta cómo las organizaciones ofrecen dichos servicios. En el caso de los grupos SQA, se trata de grupos totalmente independientes al proceso de desarrollo que centran sus esfuerzos de pruebas a un nivel más enfocado al propio proceso.

Simplemente echando un vistazo a la variedad de organizaciones que se han presentan a estudio, resulta fácil adivinar la gran cantidad de formas de acometer las pruebas que existen en la actualidad. Tantas como organizaciones quieran implementarlas.

Por ello, se establecen una serie de variables que faciliten su elección y que permitan adaptar la forma de hacer pruebas a la organización. La primera de ellas, dado el peso que supone es la propia **economía** en la empresa, representando la reducción de costes que implica una determinada organización.

Atendiendo a la adecuación de la organización a las características de un proyecto nos encontramos con:

- **El empleo de últimas tecnologías.** Éste es un concepto básico en el desarrollo de las TI, el proyecto puede evolucionar y es necesario que la organización de pruebas pueda responder a las nuevas exigencias. Se encuentra orientado a la predisposición de una organización por la formación y por la utilización de nuevas tecnologías.
- **Independencia técnica.** Una organización con independencia técnica posee una estructura técnica independiente al desarrollo, permitiendo una especialización de tanto los recursos como de las metodologías y herramientas aportando objetividad.
- **Independencia directiva.** La organización con independencia directiva permite una toma de decisiones centrada en la calidad, libre de presiones.
- **SQA:** El desarrollo de labores de SQA representa el interés de la organización no solo en la realización de las pruebas sino en la calidad del propio proceso.

Las características del personal que componen estas organizaciones es esencial para su correcto funcionamiento, aunque en este capítulo no se atiende a características individuales de los mismos, sí se hace a las de su conjunto. Por ello, es importante que el personal cumpla con:

- **Implicación por la calidad.** Un alto grado de motivación de la organización con el proyecto mejora las expectativas de calidad.

Junto a estas pautas, el que un grupo cumpla con modelos y políticas de trabajo permite a la empresa estudiar su impacto dentro de su organización.

Por último, y sin olvidar uno de los factores más importantes hoy en día, hay que considerar la capacidad de comunicación en la organización dado que su fluidez y el uso de protocolos impactan directamente en el desarrollo del proyecto, pudiendo reducir o aumentar el tiempo en la toma de decisiones.

Con el fin de realizar un análisis basado en los diferentes modos de hacer pruebas propuestos, se atiende a los siguientes factores, los cuales pueden ser claves a la hora de afrontar la elección de una organización que optimice los procesos de calidad dentro de un determinado proyecto.

Análisis de las competencias y los perfiles profesionales.

Por último, la información referente a las competencias y a los perfiles profesionales refleja cómo se recogen las distintas funciones que acarrea el desarrollo de un proceso de pruebas por parte del personal al cargo. Aunque como se puede ver a largo del mismo se trata de un concepto que actualmente no se encuentra recogido en profundidad. Por lo cual, para concluir dicho capítulo se establece una propuesta que señala las distintas labores de v&v presentadas en los capítulos anteriores y cómo podrían ser distribuidas de forma establezcan una serie de perfiles especializados.

Esta información centrada busca la manera de reflejar cómo se encuentran representados las actividades y las necesidades que implica la realización de las pruebas por parte de los profesionales del sector. Por ello, la documentación se ha basado en referencias que marcan el actual desarrollo de las mismas, como son “*los perfiles de capacidades profesionales genéricas de TIC de Career Space*” [CARRER], “*el libro blanco de ANECA*” [ANECA], el proyecto Tuning [DEUSTO00], TMap Next [SEI06] y el modelo que sigue MTP.

Teniendo en cuenta que desde la formalización de las pruebas como un proceso, éstas no son consideradas como una simple actividad. Sino que además, éstas tienen presencia a lo largo de un ciclo de vida especializado. Es necesario que el grupo de profesionales encargados de llevarlas a cabo obtenga unos conocimientos especializados.

Para poder evaluar estos conocimientos adquiridos, al igual que en los otros análisis, en éste se van a presentar una serie de características que han de presentar para su posterior evaluación:

- **Conocimientos orientados a las pruebas:** El conocimiento del entorno de las pruebas supone el desarrollo de la capacidad de adaptación de un trabajador. Es importante que el empleado conozca modelos orientados a las pruebas, la necesidad de realizarlas y las características más básicas que estas presentan en todo su proceso.
- **Conocimiento de herramientas específica:** Cada vez es mayor el número de herramientas en el mercado que facilita la realización de pruebas. Su conocimiento implica contar con un personal especializado.
- **Capacidades de planificación del proceso:** Como todo proceso, las pruebas necesitan de la planificación de recursos y actividades. Un conocimiento especializado en este orientado al desarrollo de las pruebas resultará fundamental para su correcta implantación.
- **Capacidades de análisis de casos de prueba:** Debido a que la realización de las pruebas resulta conceptualmente contraria al desarrollo, para poder encontrar fallos en un sistema es importante tener los conocimientos necesarios que permitan obtener los mejores casos de prueba acordes con las necesidades planteadas.
- **Capacidades técnicas:** La realización de pruebas resulta una tarea excesivamente especializada. Para lo cual el conocimiento de técnicas y medidas de las pruebas y el conocimiento del entorno sobre el que van a llevarse estas resulta fundamental.

3.2.1 CMMI: Modelo de capacidad y madurez integrado

A finales del siglo XX, un estudio encargado por el departamento de defensa de EE.UU. (DoD) a sus mayores contratistas reveló, que los proyectos pactados por 28 meses sufrían retrasos de otros 20 y, que un proyecto de 4 años no terminaba en menos de 7 [SCHAEFFER04]. La necesidad de solventar este hecho provocó el nacimiento de CMM (Capability Maturity Model) que, impulsado por Watts S. Humphrey, nace como un modelo de calidad de desarrollo software. Este fue el primer gran modelo de calidad que englobaría las distintas metodologías surgidas hasta la fecha, y dotaría al usuario de una evaluación sobre la madurez de sus procesos.

Años más tarde se desarrollaría CMMI [PYSTER] que, derivado de CMM, incorpora mejoras basadas en la experiencia. CMMI es un marco relativamente completo y ampliamente utilizado para la mejora de procesos en organizaciones que construyen productos complejos de ingeniería. Bajo el amparo de organizaciones como DoD, la National Defense Industrial Association (NDIA), y del Software Engineering Institute (SEI), se ha convertido en un modelo ampliamente utilizado en más de 30 países distintos tal y como señala la Web [SEI] del propio SEI. CMMI posee dos representaciones: representación por etapas y representación continua, definiendo para cada una de ellas un conjunto de niveles que engloban distintas áreas de proceso software. Éstas se encuentran basadas en criterios bien definidos que requieren una evaluación constante y rigurosa [KASSE]. Aun así, existen empresas que a pesar de tener un alto nivel de madurez no han

demostrado una línea constante de trabajo de calidad. El Sr. Mark Schaeffer, director de ingeniería de sistemas para el DoD presentó una información en la cual se demostraba que la correlación entre la calidad y los altos niveles de madurez no era tan fuerte como se creía [SCHAEFFER04].

A continuación, se describe el modelo CMMI teniendo en cuenta los diferentes elementos que lo componen y prestando especial atención a las áreas de proceso.

3.2.1.1 Estructura de CMMI

CMMI [SEI06] es un modelo de referencia que engloba el desarrollo y el mantenimiento tanto de productos como de servicios software. En él, se desarrollan prácticas que abarcan la gestión del proyecto, la gestión de procesos, la ingeniería de sistemas, la ingeniería del software, la ingeniería del hardware y otros procesos de soporte utilizados en el desarrollo y el mantenimiento. CMMi ofrece una mejora de procesos a partir de dos representaciones distintas, la representación continua y la representación por etapas. Ambas se encuentran íntimamente ligadas al desarrollo de áreas de proceso (AP) que caracterizan distintos comportamientos de la organización. Ambas representaciones comparten los mismos componentes, representados en la figura que se muestra a continuación:

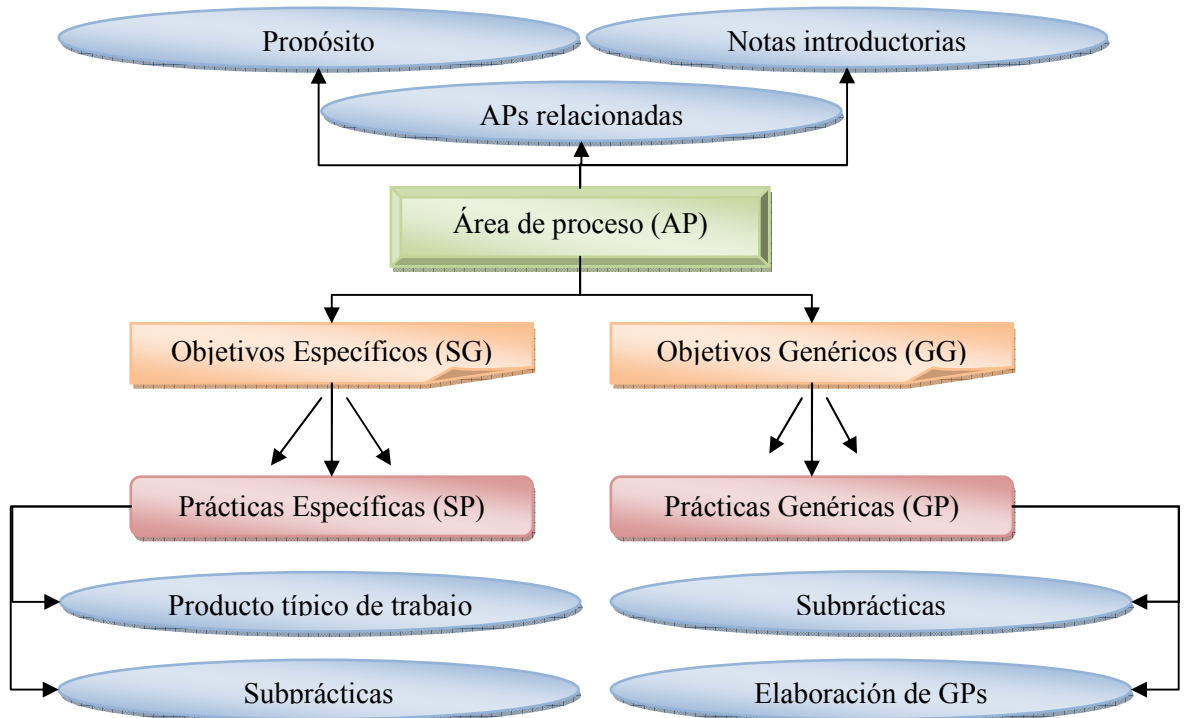


Figura 7. Estructura de los componentes de CMMI.

Las **áreas de proceso (APs)** son un conjunto de prácticas relacionadas que, cuando son implementadas colectivamente, satisfacen un conjunto de objetivos considerados importantes para conseguir una mejora en dicho área. Las áreas de proceso están formadas por componentes informativos y requeridos [SEI06].

Los “*componentes informativos*” simplemente son distintas informaciones que describen un determinado aspecto de un área de proceso, o de un cierto elemento. Se conocen como componentes informativos de un área de proceso al ***propósito***, el cual describe el objetivo del AP, a las ***notas introductorias***, que describen los conceptos abarcados por el área, y a las ***áreas de proceso relacionadas***, las cuales reflejan la relación existente entre dicho área de proceso y el resto.

Los “*componentes requeridos*” son características que son necesarias realizar para conseguir cumplir con una determinada área de proceso o un determinado elemento. Se conocen como componentes requeridos de un AP tanto a los ***objetivos específicos (SG)*** como a los ***objetivos genéricos (GG)***. Ambos, describen las características necesarias para completar un área de proceso, aunque en el caso de los objetivos genéricos estas características resultan ser comunes a un conjunto APs. Los dos tipos de objetivos están formados por una serie de elementos conocidos como componentes esperados.

Los “*componentes esperados*” son actividades que han de realizarse, aunque el conjunto de ellas no son obligatorias ni para cumplir los objetivos genéricos ni los específicos.

Los componentes esperados de los objetivos específicos son las ***prácticas específicas (SP)***. Éstas a su vez, están compuestas por una serie de componentes informativos llamados subprácticas y productos típicos de trabajo. Las ***subprácticas*** de las SPs son descripciones detalladas que ofrecen una guía para interpretar e implementar las prácticas específicas, mientras que los ***productos típicos de trabajo*** describen una lista de ejemplos de salida de una práctica específica [SEI06].

De forma similar a las prácticas específicas, los componentes esperados de los objetivos genéricos son las ***prácticas genéricas (GPs)***. También comparten una estructura similar con las prácticas específicas, al basarse en componentes informativos, los cuales en este caso son las subprácticas y las elaboraciones de GPs. Las ***subprácticas*** de las prácticas genéricas son descripciones detalladas que ofrecen una guía para interpretarlas e implementarlas, mientras que las ***elaboraciones prácticas*** se desarrollan posteriormente a las GPs, con el fin de poder establecer una guía de cómo realizar una práctica genérica [SEI06].

Representaciones

A la hora de implantar CMMi son conocidas dos representaciones distintas, la representación continua y por etapas.

La **representación por etapas** de CMMI ofrece una manera sistemática y estructurada de enfocar el modelo, el cual está basado en la mejora de procesos. Al cumplir cada etapa del mismo, se asegura el desarrollo de la infraestructura de procesos adecuada para comenzar la siguiente etapa del modelo. En la representación por etapas, se describe un orden para la implementación de áreas de proceso acorde con los niveles de madurez, definiendo así, una trayectoria de mejora para la organización, desde un nivel inicial a un nivel optimizado. Además, permite dar a conocer el nivel de madurez en el que se encuentra una determinada organización y, qué procesos hay que trabajar para obtener una mejora de los mismos [SPR].

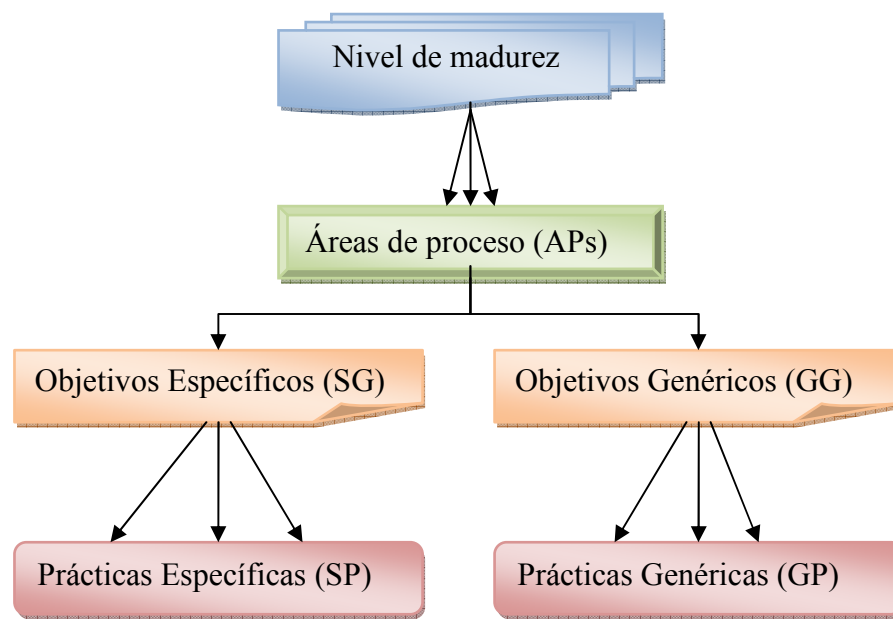


Figura 8. Estructura de la representación por etapas de CMMI.

Por otra parte, la **representación continua** ofrece una flexibilidad máxima cuando se usa CMMI para la mejora de procesos. Una organización puede elegir mejorar el rendimiento de un solo proceso o puede trabajar con distintas áreas, permitiendo mejorar diferentes procesos a diferentes niveles. Las dependencias que existen entre algunas áreas de proceso pueden, sin embargo, limitar un poco las elecciones. La representación continua se centra en las mismas áreas de proceso que la representación por niveles aunque, no evalúa la madurez de la organización sino la capacidad sus procesos [SCHIJNDEL].

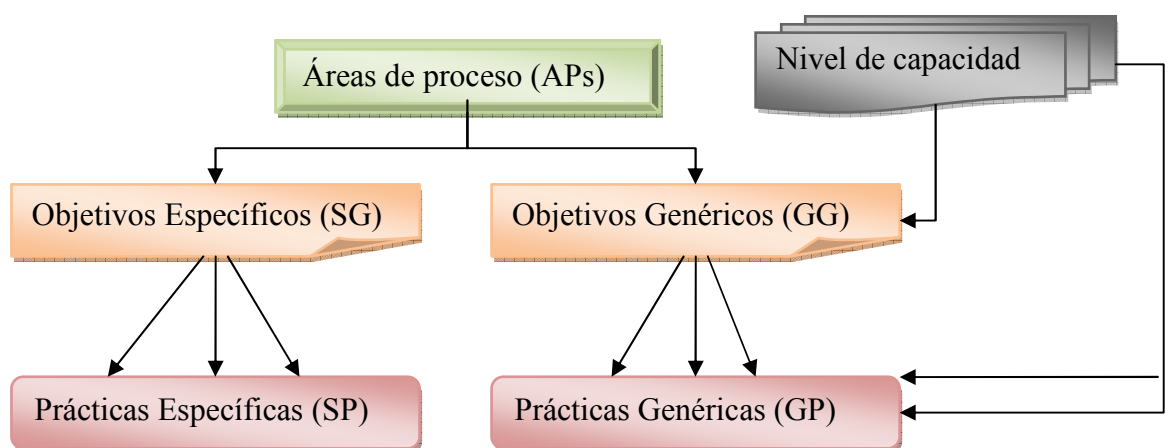


Figura 9. Estructura de la representación continua de CMMI.

Ambas representaciones poseen diferentes números de niveles, seis de capacidad en la representación continua y cinco de madurez en la representación por etapas.

Para dar soporte a la representación por etapas, CMMI presenta diferentes **niveles de madurez**. Los cuales consisten en una serie de prácticas genéricas y específicas relacionadas por un conjunto de áreas de proceso, cuya elaboración proporciona la mejora de un conjunto de procesos de la organización, y establece la infraestructura de procesos necesaria para poder alcanzar el siguiente nivel. Por este motivo, para poder alcanzar un cierto nivel de madurez, se ha de cumplir con los anteriores, ya que incluso cabe la posibilidad de continuar las áreas de proceso.

Atendiendo a la representación continua, se presentan **los niveles de capacidad** que se centran en la capacidad de las áreas de proceso. Estos niveles se aplican con el fin de mejorar de forma incremental los procesos que corresponden a un área de proceso individual. Existen seis niveles de capacidad, numerados de 0 a 5. Estos al estar referidos a una única área de proceso en cada caso, y no a un conjunto de ellas, como en el caso de la representación por etapas, los niveles de capacidad permiten una representación mucho más específica del estado de los procesos [KNEU08].

Estos dos tipos de representaciones cuentan con una relación directa entre los distintos niveles así, a partir del nivel 2 comparten las mismas características, pero centrándose la representación continua en una única área de proceso, mientras que la representación por etapas lo hace en un conjunto de áreas marcado por dicho nivel. Sin embargo, tanto el nivel 1 como el nivel 0 (éste solo presente en la representación continua), varían respecto a las dos representaciones. Este hecho viene dado precisamente por la característica que diferencia al resto de niveles, el número de áreas de proceso al que se hace referencia. Por ello, la existencia de un solo proceso realizado completa el nivel 1 de la representación continua, mientras que en la representación por etapas se hace necesaria la existencia del conjunto de áreas marcadas para completar dicho nivel. Por lo tanto, se hace necesaria la existencia del nivel 0 en la representación continua para reflejar que no existe ningún proceso realizado, mientras que este hecho no toma importancia en la representación por etapas [SEI06].

NIVEL	CAPACIDAD	MADUREZ
0	Incompleto	N/A
1	Realizado	Inicial
2	Gestionado	Gestionado
3	Definido	Definido
4	Gestionado cuantitativamente	Gestionado cuantitativamente
5	Optimizado	Optimizado

Tabla 2. Tabla de relación entre los niveles CMMI.

Existen cinco o seis niveles dependiendo de la representación, los cuales son resumidos a continuación [SCHIJNDEL] [SPR] [KNEU08]:

Nivel 0. Incompleto: Un proceso de nivel 0 se presenta como un *“proceso incompleto”* el cual, o bien no se realiza o, se realiza parcialmente. En este nivel, al menos uno de los objetivos específicos del área de proceso no se satisface. Además, no existen objetivos genéricos para dicho nivel, ya que no hay ninguna razón para institucionalizar un proceso realizado parcialmente.

Nivel 1. Realizado: Un proceso de nivel 1 se presenta como un *“proceso realizado”*, el cual satisface los objetivos específicos del área de proceso, dando soporte y permitiendo el trabajo necesario para producir los productos planificados. Aunque el nivel de capacidad 1 ofrece mejoras importantes, esas mejoras pueden perderse con el paso del tiempo si no se institucionalizan. La aplicación de la institucionalización (las prácticas genéricas de CMMI en los niveles de capacidad 2 a 5) ayuda a asegurar que las mejoras se mantienen.

Nivel 2. Gestionado: Un proceso perteneciente al nivel 2 es conocido como un *“proceso gestionado”* el cual, es un *proceso realizado* (nivel de capacidad 1) que, tiene la infraestructura básica necesaria para dar soporte al proceso y se ejecuta de acuerdo con diversas políticas. Para satisfacer estas necesidades, un proceso gestionado emplea personal con los conocimientos y las habilidades requeridos por el proyecto, e involucra a las distintas partes interesadas de forma que el proceso pueda ser controlado, seguido y evaluado en función de sus requisitos. La disciplina de proceso reflejada por el nivel de capacidad 2 ayuda a asegurar que las prácticas existentes se mantienen durante tiempos de estrés.

Nivel 3. Definido: Un proceso perteneciente al nivel 3 se conoce como un *“proceso definido”*, el cual es un *proceso gestionado* (nivel de capacidad 2) que se adapta a partir del *conjunto de* procesos estándar de la organización, de acuerdo a las guías de adaptación de la organización, y contribuye a los activos de proceso de la organización con productos, medidas e información adicional de la mejora de procesos. En un proceso claramente definido se establece su propósito, entradas, criterios de entrada, actividades, roles, medidas, etapas de verificación, productos y criterios de salida. Además en el nivel 3, los procesos son administrados con mayor proactividad que en el nivel 2 estudiando las interrelaciones entre las actividades de los procesos y detallando las medidas, productos y servicios de los mismos.

Por último, los proyectos establecen sus procesos definidos adaptando el conjunto de procesos estándar de la organización de acuerdo a las guías de adaptación. En este hecho radica una gran distinción con el nivel de madurez, ya que por el cual, en el nivel de madurez, los estándares, las descripciones de proceso y los procedimientos de un proyecto se adaptan para adecuarse a un proyecto particular o a una unidad organizativa a partir del conjunto de procesos estándar de la organización y, por tanto, son más consistentes, exceptuando las diferencias permitidas por las guías de adaptación.

Nivel 4. Gestionado cuantitativamente: Un proceso perteneciente al nivel 4 es conocido como un *“proceso gestionado cuantitativamente”*, el cual es un proceso definido (nivel de capacidad 3) controlado utilizando técnicas estadísticas y otras técnicas cuantitativas. Por ello, se establecen objetivos cuantitativos de calidad y de ejecución del proceso, y se utilizan como criterios para gestionar el proceso. Tanto la calidad como el rendimiento de los procesos se estudian en términos estadísticos y son gestionados durante toda la vida de los procesos.

Nivel 5. Optimizado: Un proceso de nivel 5 se caracteriza como un *“proceso optimizado”*, el cual es un *proceso gestionado cuantitativamente* (nivel de capacidad 4) mejorado en base a una comprensión de las causas comunes de variación inherentes al proceso. El enfoque de un proceso optimizado es mejorar continuamente la tasa de rendimiento del proceso mediante mejoras, tanto incrementales como innovadoras.

Una vez definidos los niveles y tal y como se ha mencionado con anterioridad, a la hora de implantar una representación por etapas, la consecución de un nivel de madurez se obtiene gracias a la elaboración de distintas áreas de proceso, cuya relación puede ser vista a continuación:

NIVEL DE MADUREZ	AP	
OPTIMIZADO	Innovación y despliegue en la organización. Análisis de causas y resolución.	OID CAR
GESTIONADO CUANTITATIVAMENTE	Rendimiento de procesos de la organización Gestión cuantitativa del proyecto.	OPP QPM
DEFINIDO	Definición de proceso de la organización. Enfoque en procesos de la organización. Formación organizativa. Gestión integrada de proyecto. Gestión de riesgos. Desarrollo de requerimientos. Solución técnica. Integración de producto. Verificación. Validación. Análisis de decisión y resolución.	OPD OPF OT IPM RSKM RD TS PI VER VAL DAR
GESTIONADO	Planificación del proyecto. Supervisión y control del proyecto. Gestión de contratos de proveedores. Gestión de requerimientos. Gestión de la configuración. Aseguramiento de la calidad de proceso y de producto. Medición y análisis.	PP PMC SAM REQM CM PPQA MA
INICIAL	No tiene.	-

Tabla 3. Tabla de relación de niveles de madurez y APs en CMMi.

3.2.1.2 Áreas de proceso

Las **áreas de proceso (APs)** son unos de los principales elementos estructurales de CMMi [KNEU08]. Están compuestas por un conjunto de prácticas relacionadas (véase la Figura 7), que al completarse en conjunto, satisfacen los objetivos establecidos para su mejora.

En CMMi existen veintidós áreas de proceso, divididas en cuatro categorías, aunque estas categorías tienen un significado meramente estructural, no teniendo efecto en la implementación práctica del modelo [KNEU08]:

Gestión de procesos. Las áreas proceso de gestión de procesos contienen las actividades transversales a los proyectos relacionadas con la definición, planificación, despliegue, implementación, monitorización, control, evaluación, medición y mejora de los procesos:

- Definición de la organización de los procesos.
- Formación de la organización.
- Realización de los procesos de organización.
- Innovación organizativa y de implementación.

Gestión del proyecto. Las áreas proceso de gestión de proyectos cubren las actividades de gestión de proyectos relacionadas con: la planificación, la monitorización y el control de los proyectos:

- Planificación del proyecto.
- Supervisión y control del proyecto.
- Gestión de contratos con los proveedores.
- Gestión del proyecto integrado.
- Gestión de riesgos.
- Gestión cuantitativa del proyecto.

Ingeniería. Las áreas proceso de ingeniería cubren las actividades de desarrollo y de mantenimiento que se comparten entre las disciplinas de ingeniería. Integran procesos asociados con diferentes disciplinas de ingeniería en un único proceso de desarrollo del producto, con lo que se da soporte a una estrategia de mejora de procesos orientada a producto. Estos procesos se aplican al desarrollo de cualquier producto o servicio dentro del dominio de desarrollo:

- Gestión de requisitos.
- Desarrollo de los requisitos.
- Solución técnica.
- Integración de producto.
- Verificación.
- Validación.

Soporte. Las áreas de proceso de soporte cubren las actividades que, como su propio nombre indica, dan soporte al desarrollo y al mantenimiento del producto. Estas áreas tratan los procesos relacionados con el entorno en el que se desarrollan el resto de procesos de un proyecto:

- Gestión de la configuración.

- Garantía de la calidad del producto y proceso.
- Análisis y medida.
- Análisis de decisión y de resolución.
- Análisis de causas y de resolución.

A lo largo de estas áreas de proceso se distinguen diferentes componentes, unos **informativos** que proporcionan detalles que ayudan a las organizaciones a comenzar a pensar en cómo aproximarse a los resultados, y los **requeridos** que describen lo que una organización debe realizar para satisfacer un área de proceso. Los primeros se componen por:

- El **propósito**, que describe la finalidad del área de proceso.
- Las **notas introductorias**, que describen los conceptos principales cubiertos por el área de proceso.
- Las **áreas de proceso relacionadas**, que referencia y refleja las relaciones de alto nivel existentes entre un determinado áreas de proceso y el resto.

Sin embargo, los componentes requeridos se encuentran más formalmente definidos ya que estos logros se deben implementar de forma visible en los procesos de una organización. Utilizándose la satisfacción de los objetivos en las evaluaciones como base para, determinar si un área de proceso ha sido realizada y satisfecha [KNEU08] [PERSSE06] [DENNIS03].

Dentro de las veintidós áreas de proceso existentes en CMMI, existen dos que toman una especial relevancia dado el tema que centra este proyecto fin de carrera. Estas áreas y sus respectivos objetivos y prácticas específicas son:

Verificación: El propósito de la *verificación (VER)* es el de asegurar que los productos seleccionados cumplen los requerimientos especificados. Es un área de proceso de ingeniería en el nivel de madurez 3.

Objetivos y prácticas específicas.

- **SG1.** Preparación para la verificación.
 - **SP1.1.** Establecer entorno de verificación.
 - **SP1.2.** Establecer verificación de procedimientos.
 - **SP1.3.** Seleccionar productos para la verificación.
- **SG2.** Realización de revisión de pares.
 - **SP2.1.** Analizar revisión de pares.
 - **SP2.2.** Guiar revisión de pares.
 - **SP2.3.** Preparar para la revisión de pares.

- **SG3.** Verificación de productos seleccionados.
 - **SP3.1.** Analizar resultados.
 - **SP3.2.** Realizar verificación.

Validación: El propósito de la *validación (VAL)* es el de demostrar que un producto o un componente del mismo se ajusta a su uso previsto cuando se sitúa en su entorno previsto. Es un área de proceso de ingeniería en el nivel de madurez 3.

Objetivos y prácticas específicas.

- **SG1.** Preparación de la validación.
 - **SP1.1.** Establecer entorno de validación.
 - **SP1.2.** Establecer procedimiento de validación.
 - **SP1.3.** Seleccionar productos para la validación.
- **SG2.** Validación del producto o de sus componentes.
 - **SP2.1.** Analizar resultados.
 - **SP2.2.** Realizar validación.

A parte de los distintos objetivos y prácticas específicas que pertenecen exclusivamente a un área de proceso, existen los **objetivos genéricos (GG)** que son denominados precisamente como “genéricos” ya que la misma declaración del objetivo es común a distintas áreas de proceso. Un objetivo global describe las características que deben estar presentes para institucionalizar los procesos que implementan un área de proceso. Mientras que un **objetivo específico (SG)** describe las características únicas que se deben de satisfacer en un determinado área de proceso.

En la representación continua están presentes los cinco objetivos genéricos (GG) definidos en CMMI. Por el contrario, en la representación por etapas sólo se usan las metas genéricas 2 y 3. Esto sucede porque no todos los procesos pertenecientes a un determinado nivel tendrán porque estar situados por encima un proceso definido. En la siguiente tabla se relacionan los objetivos genéricos con los niveles de madurez y, por lo tanto, con las correspondientes áreas de proceso reflejadas en los distintos niveles [KNEU08].

NIVELES/OBJETIVOS	GG1	GG2	GG3	GG4	GG5
Nivel de madurez 2		X			
Nivel de madurez 3			X		
Nivel de madurez 4			X		
Nivel de madurez 5			X		

Tabla 4. Relación entre niveles de madurez y objetivos genéricos.

Por último, se distinguen los **componentes esperados** que describen lo que una organización puede implementar para lograr un componente requerido, sirviendo de guía a quien implementa las mejoras o las evaluaciones. Los componentes esperados incluyen las prácticas específicas y prácticas genéricas, las cuales deberán estar presentes en los procesos planificados e implementados para que los objetivos puedan considerarse satisfechos [SEI06].

Las **prácticas genéricas** al igual que los objetivos genéricos se aplican a múltiples áreas de proceso. Una práctica genérica es la descripción de una actividad que se considera importante para el logro de un objetivo genérico asociada.

Las **prácticas específicas (SP)** son descripciones de las actividades que se consideran importantes para conseguir un determinado objetivo específico [SEI06].

A continuación, se describen los cinco objetivos genéricos existentes en CMMI y sus correspondientes prácticas genéricas [SEI06]:

GG1: Conseguir los objetivos específicos.

El proceso da soporte y permite el logro de los objetivos específicos del área de proceso. Para ello transforma los productos entrada productos de salida.

- **GP1.1 Realizar prácticas específicas.** Realizar las prácticas específicas del área de proceso para desarrollar los productos de trabajo y proporcionar servicios para lograr las metas específicas del área de proceso. El propósito de esta práctica genérica es producir los productos y entregar los servicios que se esperan al realizar el proceso.

GG2: Institucionalizar el proceso gestionado.

El proceso se encuentra institucionalizado como un proceso gestionado.

- **GP2.1. Establecer una política organizacional.** *Establecer y mantener una política de la organización para planificar y realizar el proceso.* El propósito de esta práctica genérica consiste en definir las expectativas de la organización en relación con el proceso, y hacerlas visibles para aquellos que se vean afectados dentro de la organización.
- **GP2.2. Plan de procesos.** *Establecer y mantener el plan para realizar el proceso.* El propósito de esta práctica genérica es el de determinar lo que se necesita para realizar el proceso y para lograr los objetivos establecidos. Para ello, se prepara y se acuerda con las partes interesadas, un plan y una descripción para realizar el proceso.
- **GP2.3. Proveer recursos.** *Proporcionar recursos adecuados para realizar el proceso, desarrollar los productos de trabajo y proporcionar los servicios del proceso.* El propósito de esta práctica genérica es el de asegurar la disponibilidad de los recursos necesarios para realizar el proceso, tal y como se definieron en el plan. Los recursos incluyen: la financiación adecuada, instalaciones físicas apropiadas, el personal cualificado y las herramientas apropiadas.
- **GP2.4. Asignar responsabilidades.** *Asignar la responsabilidad y la autoridad para realizar el proceso, desarrollar los productos de trabajo y*

proporcionar los servicios del proceso. El propósito de esta práctica genérica es el de asegurar que existe responsabilidad para realizar el proceso y para lograr los resultados especificados a lo largo de la vida del proceso. Las personas asignadas deben tener la autoridad apropiada para realizar las responsabilidades que les son asignadas.

- **GP2.5. Formación de personal.** *Formar, según sea necesario, a las personas para realizar o dar soporte al proceso.* El propósito de esta práctica genérica es asegurar que las personas posean las habilidades y la experiencia necesaria para realizar o para dar soporte al proceso.
- **GP2.6. Administrar las configuraciones.** *Poner los productos de trabajo designados del proceso bajo los niveles de control apropiados.* El propósito de esta práctica genérica es el de establecer y mantener la integridad de los productos del proceso a lo largo de su vida útil.
- **GP2.7. Identificar e involucrar las partes interesadas importantes.** *Identificar e involucrar, cómo se planificó a las partes interesadas relevantes del proceso.* El propósito de esta práctica genérica es el de establecer y mantener la implicación prevista de las partes interesadas durante la ejecución del proceso.
- **GP2.8. Supervisar y controlar el proceso.** *Seguir y controlar el proceso frente al plan para realizar el proceso y tomar las acciones correctivas apropiadas.* El propósito de esta práctica genérica es el de realizar el seguimiento y el control directo del proceso día a día.
- **GP2.9. Evaluar objetivamente su aplicación.** *Evaluar objetivamente la adherencia del proceso frente a la descripción del proceso, estándares y procedimientos, y tratar las no conformidades.* El propósito de esta práctica genérica es el de garantizar objetivamente que el proceso se implementó según lo planificado y, que se adhiere a la descripción del proceso, de los estándares y de los procedimientos.
- **GP2.10. Revisar la situación con un mayor nivel de gestión.** *Revisar con el nivel directivo las actividades, el estado y los resultados del proceso, y resolver los problemas.* El propósito de esta práctica genérica es el de proporcionar la visibilidad apropiada del proceso a nivel directivo.

GG3: Institucionalizar el proceso definido.

El proceso se encuentra institucionalizado como un proceso definido.

- **GP3.1. Establecer un proceso definido.** *Establecer y mantener la descripción de un proceso definido.* El propósito de esta práctica genérica es el de establecer y mantener una descripción del proceso. La cual, se adapte a partir del *conjunto de procesos estándar de la organización*¹ con el fin de tratar las necesidades de una instanciación específica.

- **GP3.2. Recoger información de mejora.** *Recoger productos de trabajo, medidas, resultados de medición e información de mejora procedente de la planificación y realización del proceso para dar soporte al uso futuro y a la mejora de los procesos y de los activos de proceso de la organización.* El propósito de esta práctica genérica consiste en recoger la información y los productos derivados de la planificación y de realización del proceso.

GG4: Institucionalizar el proceso gestionado cuantitativamente.

El proceso está institucionalizado como un proceso gestionado cuantitativamente.

- **GP4.1. Establecer objetivos cuantitativos para el proceso.** Establecer y mantener los objetivos cuantitativos para el proceso, los cuales tratan la calidad y el rendimiento del proceso en base a las necesidades del cliente y a los objetivos de negocio. El propósito de esta práctica genérica consiste en determinar y obtener un acuerdo con las partes interesadas relevantes sobre los objetivos cuantitativos y específicos del proceso.
- **GP4.2. Estabiliza el rendimiento de subprocesos.** Estabilizar el rendimiento de uno o más subprocesos para determinar la capacidad del proceso para lograr los objetivos cuantitativos establecidos de calidad y de rendimiento del proceso. El propósito de esta práctica genérica consiste en estabilizar el rendimiento de uno o varios subprocesos del propio proceso definido. Los cuales contribuyen de manera crítica al rendimiento general mismo, utilizando técnicas estadísticas y otras técnicas cuantitativas apropiadas.

GG5. Institucionalizar un proceso optimizado.

El proceso está institucionalizado como un proceso en optimización.

- **GP5.1. Asegurar la mejora continua del proceso.** *Asegurar la mejora continua del proceso para satisfacer los objetivos de negocio relevantes de la organización.* El propósito de esta práctica genérica consiste en seleccionar y aplicar sistemáticamente las mejoras de los procesos y de la tecnología. De tal forma, que contribuyan a cumplir los objetivos establecidos de calidad y de rendimiento del proceso.
- **GP5.2 Causa correcta de los problemas.** *Identificar y corregir la raíz de las causas de los defectos y de otros problemas en el proceso.* El propósito de esta práctica genérica consiste en analizar los defectos y otros problemas localizados en un proceso gestionado cuantitativamente. Esto se realiza con el fin de corregir la raíz de las causas de los defectos y los problemas, y para prevenir que estos mismos no vuelvan a producirse en un futuro.

3.2.2 TMM: Modelo de madurez de las pruebas

El modelo de madurez de las pruebas (Testing Maturity Model, TMM), fue desarrollado por el Instituto de Tecnología de Illinois, con el propósito de servir como una guía para realizar una mejora del proceso de pruebas y servir como complemento a CMM. Al igual que éste, TMM también utiliza el concepto de niveles de madurez para la evaluación y la mejora de los procesos.

Con el fin de definir los niveles de madurez y como punto de partida utiliza el modelo de la evolución de las pruebas (Gelperin y Hetzel, 1988) [VEENENDAAL06]. Este modelo refleja las fases que se deben de desarrollar en las pruebas de una organización, desde un período de depuración y orientado a la detección de defectos, a un período de prevención de los mismos. Por otra parte, varias de las mejores prácticas industriales también han contribuido al desarrollo de TMM, dotándolo así de la funcionalidad y de la práctica necesaria.

En TMM se definen y se utilizan conceptos fundamentales de: dirección, medición, supervisión y eficacia. Estos conceptos son empleados con el fin de probar madurez de los procesos [SWINKELS]. Además, define pruebas que abarcan todo el proceso de calidad del software. Y conserva la misma idea de CMM de implementar cinco niveles representativos de la madurez de procesos, permitiendo a las organizaciones migrar sus procesos inmaduros a procesos probados y maduros. La estructuración de este modelo facilita la aplicación de las pruebas, delimitando su contexto, su funcionalidad y sus objetivos. Esto posibilita una mejora de la eficiencia en las pruebas [SWAN96].

3.2.2.1 Estructura de TMM

TMM se presenta como un conjunto de **niveles de madurez** que definen una jerarquía de madurez de pruebas. Cada nivel representa un escenario en la evolución hacia un proceso de pruebas maduro. La Figura 10 [BUMSTEIN03] muestra a continuación la estructura de TMM:

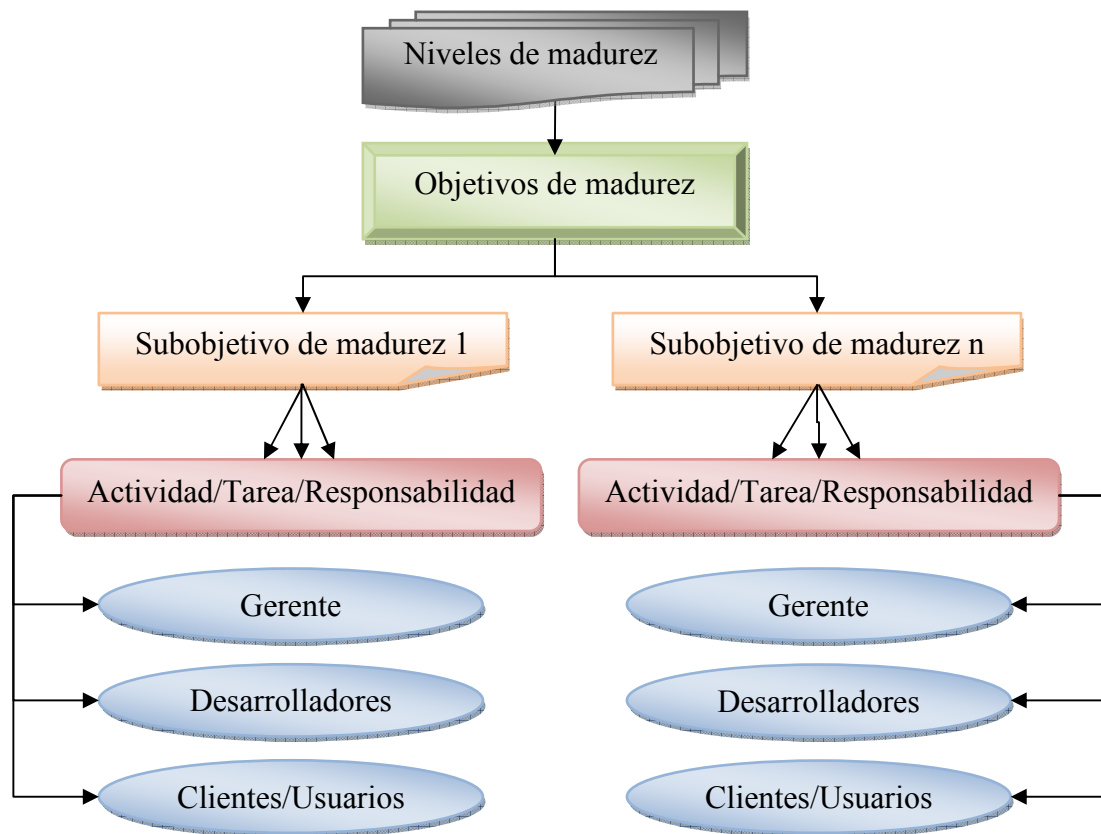


Figura 10. Estructura de TMM.

A excepción del primer nivel de TMM, en cada nivel aparecen representados distintos **objetivos de madurez** [BUMSTEIN03], los cuales identifican las distintas áreas claves del proceso. Estos objetivos establecen el alcance de la mejora de las pruebas que debe abordarse con el fin de alcanzar la madurez esperada en un determinado nivel. Para ser situada en cierto nivel, una organización debe satisfacer todos los objetivos de madurez del mismo.

A cada objetivo le prestan soporte distintos **subobjetivos madurez** [BUMSTEIN03]. Los cuales resultan menos abstractos, y definen el alcance, los límites, y los logros necesarios para alcanzar un nivel determinado. Los subobjetivos madurez se logran a través del cumplimiento de un grupo de actividades, tareas y responsabilidades.

Las **actividades y las tareas** [BUMSTEIN03] definen las acciones que deben realizarse en un determinado nivel para mejorar la capacidad de las pruebas, estando vinculadas a los compromisos de la organización. Las **responsabilidades** [BUMSTEIN03] sin embargo, son las asignaciones de dichas actividades y tareas a los grupos clave que se consideran que representan un determinado proceso de pruebas: los gerentes, los desarrolladores y probadores, y los usuarios y los clientes. La definición de sus roles es esencial en el desarrollo de un marco de madurez.

El **gerente** [BUMSTEIN03] implica el compromiso y la capacidad para realizar actividades y tareas para mejorar la capacidad de las pruebas.

El *desarrollador y el probador* [BUMSTEIN03] abarcan las actividades técnicas y las tareas que, cuando se aplican, constituyen prácticas de pruebas maduras.

Los *usuarios y los clientes* [BUMSTEIN03] ofrecen un punto de apoyo al proceso de pruebas. Para ello, se solicita su aprobación y participación en actividades tales como análisis de requerimientos, pruebas de uso y planificación de las pruebas de aceptación.

Representación

TMM cuenta con cinco niveles de madurez. Un nivel de madurez en TMM se entiende como el grado de calidad de los procesos de pruebas de la organización [VEENENDAAL06]. Cada nivel de madurez consiste en unos determinados objetivos de madurez. Éstos indican donde la organización debe actuar para mejorar su proceso de pruebas, a excepción de en el nivel 1 que carece de áreas de proceso. Para alcanzar un determinado nivel, previamente se tienen que alcanzar todos los objetivos de madurez del propio nivel y de su predecesor. Los niveles de TMM se definen de la siguiente forma [VEENENDAAL06] [SWINKELS] [DRIEL]:

Nivel 1. Inicial: En el primer nivel de madurez, el proceso de pruebas es caótico e indefinido pero se considera como parte fundamental en la resolución de errores. El objetivo fundamental en este nivel consiste en demostrar que el software funciona sin fallos importantes. Por ello, a menudo los productos se lanzan sin la visibilidad adecuada acerca en cuestiones de calidad y riesgos, dando como resultado un software que no cumple con las necesidades, inestable o demasiado lento para trabajar con él. Además en este tipo de proyectos existen carencias de recursos, herramientas y probadores con formación.

Nivel 2. Definición: En el segundo nivel de madurez, el proceso de pruebas se define y se separa claramente para evitar errores. Además, para ir derivando y seleccionando los casos de prueba de las especificaciones de requisitos, se aplican técnicas de diseño formales de pruebas. El principal objetivo de las pruebas de este nivel consiste en verificar que el software satisface la especificación de requisitos. Sin embargo, todavía el proceso de pruebas comienza relativamente tarde en el ciclo vida del desarrollo [VEENENDAAL06].

Nivel 3. Integración: En el tercer nivel de madurez, el proceso de pruebas se integra completamente en el ciclo de vida del software y se reconocen en todos los niveles del modelo V. Por ello, la planificación de las pruebas se realiza en una etapa temprana del proyecto, por medio de un plan de pruebas principal. En este nivel, las pruebas son determinadas usando técnicas de gestión de riesgos y se basan en la estrategia seguida en el estudio de requisitos. Además, se realizan revisiones, aunque no constantemente y sin seguir un procedimiento documentado, y se verifica que el software satisfaga los requisitos. Todo esto conlleva a la formación de una organización de pruebas, así como a un programa de formación de pruebas, percibiéndose de esta manera el proceso de pruebas de forma profesional.

Nivel 4. Gestión y medida: En el cuarto nivel de madurez, el proceso de pruebas se encuentra bien definido y medido. En él, las revisiones y la inspección se desarrollan durante todo el ciclo de vida y el software se evalúa usando criterios de calidad de las

características, tales como la fiabilidad, la utilidad y la capacidad de mantenimiento. Los casos de pruebas se reúnen, se almacenan y se gestionan en una base de datos central para la reutilización y la revisión de pruebas. Además, un programa de medición de pruebas dota de información y visibilidad del proceso de pruebas y de la calidad del producto.

Nivel 5. Optimización: En el quinto nivel de madurez, debido a la infraestructura proporcionada por el logro de los objetivos de madurez de los niveles anteriores, el proceso de pruebas es totalmente definido y, se es capaz de controlar los costes y la efectividad de las pruebas. En éste nivel se optimizan técnicas y métodos, existiendo un enfoque continuo en la mejora del proceso de pruebas.

NIVEL DE MADUREZ	Objetivos de madurez
5 OPTIMIZACIÓN	Aplicación de datos de proceso para la prevención de defectos. Optimización del proceso de pruebas. Control de calidad.
4 GESTION Y MEDIDA	Establecer el programa de revisiones en la organización. Establecer el programa de medición de pruebas. Evaluación de la calidad del software.
3 INTEGRACIÓN	Establecer la organización de pruebas. Establecer el programa de formación en pruebas. Integración de las pruebas en el ciclo de vida. Control y seguimiento del proceso de pruebas.
2 FASE DE DEFINICIÓN	Desarrollar los objetivos de las pruebas y la depuración. Iniciar el proceso de planificación de las pruebas. Institucionalizar métodos y técnicas básicos de prueba.
1 INICIAL	No tiene.

Tabla 5. Tabla de relación de niveles y objetivos de madurez en TMM.

3.2.2.2 Objetivos de madurez

Los *objetivos de madurez (MG)* identifican las funcionalidades y el número de pruebas y actividades que se deben tratar para alcanzar un cierto nivel de madurez [SWINKELS]. En TMM, solamente se identifican los objetivos de madurez que son considerados determinantes para la capacidad del proceso de pruebas. Cada objetivo de madurez está formado por uno o más subobjetivos de madurez. Estos *subobjetivos de madurez (SO)* especifican los objetivos menos abstractos y definen objetivos, límites y necesidades para completar un nivel en particular. Un subobjetivo de madurez se compone por un conjunto de *actividad/tarea/responsabilidad (ATR)* [DAUGH02]. Las cuales dirigen la organización y la implementación hacia un determinado nivel. Las actividades y las tareas se definen en términos de acciones que deben de ser realizadas para dar nivel o mejorar la capacidad de las pruebas [SWINKELS].

En TMM existen un total de trece objetivos de madurez con sus respectivos subobjetivos, los cuales se describen a continuación:

Desarrollar los objetivos de las pruebas y la depuración.

La organización debe distinguir entre los procesos de pruebas y los procesos de depuración. Además, se deben identificar los objetivos, tareas, actividades y herramientas a utilizar, y asignar responsabilidades. Para ello, la gerencia debe establecer planes y políticas con el fin de acomodar e institucionalizar ambos procesos. La separación de ambos procesos es esencial para el desarrollo de la madurez de las pruebas ya que poseen diferentes objetivos, métodos y psicologías. Las pruebas en este nivel son actividades planificadas y pueden ser administradas, mientras que la depuración de errores no.

Subobjetivos.

- **SO1** Formar un comité de pruebas y depuración.
- **SO2.** El comité debe desarrollar y registrar los objetivos de las pruebas.
- **SO3.** El comité debe desarrollar y registrar los objetivos de depuración.
- **SO4.** La documentación de los objetivos de las pruebas y de la depuración debe ser distribuida a los directores de proyectos y a los desarrolladores.
- **SO5.** Los objetivos de las pruebas deben estar reflejados en el plan de pruebas.

Iniciar el proceso de planificación de las pruebas.

La planificación es esencial para que un proceso sea repetitivo, definido y gestionado. Ésta implica: establecer objetivos, analizar riesgos, esbozar estrategias y desarrollar especificaciones de diseño de pruebas y casos de prueba. Además, el plan de pruebas debe abordar la asignación de los recursos y las responsabilidades para las pruebas de unidad, integración, sistema y los niveles de aceptación.

Subobjetivos.

- **SO1.** El comité de pruebas debe estar consolidado.
- **SO2.** El marco de trabajo para la política de planificación de pruebas debe estar establecido y soportado por la dirección.
- **SO3.** Una plantilla de plan de pruebas debe ser desarrollada, registrada, y distribuida para los directores de proyecto y desarrolladores.
- **SO4.** Incluir los requisitos de usuario desarrollados como entrada al plan de pruebas.
- **SO5.** Las herramientas de planificación deben ser evaluadas, recomendadas y adquiridas. El uso de las mismas debe estar soportado por la dirección.

Institucionalizar técnicas y métodos de prueba básicas.

Las técnicas y métodos básicos de pruebas deben ser aplicadas a toda la organización para mejorar la capacidad del proceso de pruebas. Por ello, es importante especificar claramente dónde y cómo estas técnicas y métodos van a ser utilizados.

Subobjetivos.

- **SO1.** La dirección debe establecer un conjunto de políticas que garanticen la aplicación de las técnicas y los métodos recomendados de forma coherente en toda la organización.
- **SO2.** Formar un grupo especializado en tecnología de las pruebas para estudiar, evaluar y recomendar un conjunto de técnicas y métodos básicos de verificación y, un conjunto de herramientas sencillas de apoyo.

Establecer una organización de pruebas software.

Un organización de pruebas software se crea para identificar a un grupo de personas que van a adquirir responsabilidades sobre el proceso de pruebas. Dado que las pruebas, en su sentido más amplio, tienen una gran influencia sobre la calidad del producto y, ya que las pruebas consisten en actividades complejas, generalmente realizadas en virtud de un calendario muy apretado y con alta presión. Es necesario contar con un grupo de especialistas a cargo de este proceso bien formados y con dedicación. El grupo de pruebas es el responsable de: la planificación de las pruebas, el registro y ejecución de pruebas, la utilización de estándares, métricas y bases de datos de las pruebas, y de la reutilización, seguimiento y evaluación de las pruebas.

Subobjetivos.

- **SO1.** Establecer un grupo de pruebas con un líder, apoyos, y consolidado desde la alta dirección.
- **SO2.** Definir los roles y responsabilidades para el grupo de pruebas.
- **SO3.** Entrenar y motivar a los miembros del equipo.
- **SO4.** El grupo de pruebas establecerá la comunicación con el usuario y/o el cliente con el fin de solicitar su participación en las actividades de prueba y de colaboración en las tareas de: recolección, documentación e incorporación de las necesidades del usuario en el proceso de pruebas.

Establecer un programa de formación en pruebas.

Un programa de formación técnica garantiza el personal cualificado para formar el grupo de pruebas. Los probadores tienen que estar debidamente formados, de tal forma que puedan desarrollar su trabajo de manera eficaz y eficientemente. En el nivel 3 de TMM, el personal se forma en: la planificación, los métodos, los estándares, las técnicas y las herramientas de pruebas. En niveles más altos de TMM también deben aprender cómo definir, reunir, analizar y aplicar técnicas de pruebas relacionadas. Además, el programa de formación prepara al personal para la revisión de proceso, forma líderes de revisión e institucionaliza canales para la participación de los usuarios en el proceso de pruebas y de revisión. Esta formación incluye: cursos, estudio por cuenta del personal, seguimiento de programa y apoyo para la asistencia a los centros de formación.

Subobjetivos.

- **SO1.** La dirección debe establecer un programa de entrenamiento organizacional, proporcionando la consolidación y el soporte adecuado.
- **SO2.** Desarrollar los objetivos del entrenamiento y los planes.
- **SO3** El grupo de entrenamiento debe ser establecido, con herramientas, facilidades y materiales.

Integración las pruebas en el ciclo de vida del software.

Los técnicos y gerentes de calidad reconocen que un proceso maduro de pruebas y, un producto software de calidad necesitan que las actividades de pruebas sean desarrolladas en paralelo en todas las fases del ciclo de vida. Por ello, la planificación de las pruebas se realizará al inicio del ciclo de vida utilizando una variación del modelo en V y el personal de pruebas actuara siempre y cuando sea necesario a través de canales establecidos en las distintas fases de pruebas.

Subobjetivos.

- **SO1.** La fase de pruebas se debe partir en subfases que puedan ser integradas en el ciclo de vida del software.
- **SO2.** Desarrollar y adoptar una versión institucionalizada del modelo en V basada en la definición de subfases de prueba.
- **SO3.** La definición de estándares debe ser desarrollada para los productos de trabajo relacionados con las pruebas.
- **SO4.** Se debe establecer un mecanismo que permita al personal de pruebas trabajar con desarrolladores, facilitando la integración de la actividad de pruebas.

Controlar y seguimiento de las pruebas.

La gestión del proceso de pruebas se basa en cinco actividades principales: la planificación, la dirección, la gestión de personal, el control y la organización. En el nivel 2 de TMM se introducen la planificación, la gestión del personal y la dirección de las capacidades del proceso. Ya en el nivel 3, se introducen distintas actividades de control y seguimiento. Esto proporciona una visibilidad y garantiza que el proceso de pruebas se desarrolle según lo previsto, ya que si una actividad se desvía de su planificación, la dirección es capaz de tomar las acciones necesarias para corregir dicha desviación. El seguimiento y el progreso se consiguen gracias a la comparación de resultados con la planificación de los mismos. El soporte para el control y el seguimiento proviene de los estándares de las pruebas, los hitos, logs, planes de contingencia relacionados con las pruebas y métricas de pruebas que pueden ser utilizadas para evaluar el progreso y efectividad de las pruebas.

Subobjetivos.

- **SO1.** La organización debe desarrollar mecanismos y políticas para seguir y controlar el proceso de pruebas.

- **SO2.** Se debe definir, registrar y distribuir un conjunto básico de medidas relacionadas con el proceso de pruebas.
- **SO3.** Se debe desarrollar, registrar y documentar un conjunto de acciones correctivas y planes de contingencia, para utilizar cuando se produzcan desviaciones significativas con respecto al plan.

Establecer un programa de revisiones en la organización.

Una vez integradas las actividades de pruebas en el ciclo de vida del software en el nivel 3, en el nivel 4 esta integración es ampliada por el establecimiento de un programa de revisiones. Las revisiones se realizan en todas las fases para identificar, catalogar y eliminar defectos de los productos software y de pruebas tan pronto y efectivamente como sea posible.

Subobjetivos.

- **SO1.** La alta dirección debe desarrollar políticas de revisión, dar soporte al proceso de revisión y tomar responsabilidad para integrarlo todo en la cultura de la organización.
- **SO2.** El grupo de pruebas y el grupo de aseguramiento de la calidad del software deben desarrollar y documentar las metas, planes, procedimientos de seguimiento y mecanismos de registro para las revisiones durante todo el ciclo de vida.
- **SO3.** Los elementos de revisión deben ser especificados por la alta dirección.
- **SO4.** El personal debe ser formado para que puedan entender y seguir las políticas, prácticas y procedimientos de revisión.

Establecer un programa de medición de pruebas.

Un programa de medición de pruebas es esencial en la evaluación de la calidad del proceso de pruebas y para la mejora del mismo. Las medidas de las pruebas son vitales en seguimiento y control del proceso de pruebas. Un programa de medición de pruebas debe ser definido y gestionado cuidadosamente, especificando datos de medida para cada fase del ciclo de vida. Se utilizan medidas que estén relacionadas con el progreso, coste, datos de error y defectos de las pruebas y medidas de producto como la fiabilidad del producto.

Subobjetivos.

- **SO1.** Definir las políticas y metas de medición de pruebas.
- **SO2.** Desarrollar un plan de medición de pruebas con mecanismos para colección de datos, análisis y aplicación.
- **SO3.** Desarrollar y documentar planes de acción que apliquen resultados medidos para mejorar el proceso de pruebas.

Evaluación de la calidad del software.

Uno de los propósitos de la evaluación de la calidad del software en este nivel de TMM, se trata de relacionar los problemas de calidad del software con la adaptación del proceso de pruebas. La evaluación de la calidad del software implica definir atributos de calidad, que sean medibles, así como metas de calidad, que permitan evaluar el software. Los objetivos de calidad están vinculados a las pruebas debido a que el proceso de madurez de las pruebas debe conducir a que un software sea correcto, fiable, útil, mantenible, portátil y seguro.

Subobjetivos.

- **SO1.** La dirección, el grupo de pruebas y el grupo de aseguramiento de la calidad del software deben definir las políticas relacionadas con la calidad, las metas de calidad, y los atributos de calidad para los productos software.
- **SO2.** El proceso de pruebas debe ser estructurado, medido y evaluado para asegurar que las metas de calidad puedan ser alcanzadas.

Aplicación de datos del proceso en la prevención de defectos.

Las organizaciones maduras son capaces de aprender de sus propios errores, siguiendo esta filosofía las organizaciones con altos niveles de TMM registran defectos, analizan sus causas y sus patrones. En el nivel 5 de TMM, la prevención de defectos de aplica en todos los proyectos de la organización, siendo el equipo de prevención el responsable de las actividades de prevención de defectos interactuar con los desarrolladores para aplicar la prevención de defectos.

Subobjetivos.

- **SO1.** La dirección establecerá y dará soporte al equipo de prevención de defectos.
- **SO2.** Los defectos inyectados o eliminados deben ser identificados y registrados durante cada fase del ciclo de vida.
- **SO3.** Establecer mecanismos de análisis de causas para identificar las causas raíz de los defectos.
- **SO4.** Desarrollar planes de acción mediante la interacción de los directivos, desarrolladores y personal de pruebas para prevenir que los defectos identificados vuelvan a surgir.

Control de calidad

El control de la calidad en el nivel de madurez 4 se centra en las pruebas realizadas por grupos relacionados con tareas de calidad, que se encargan de la corrección, la seguridad, la portabilidad, la interoperabilidad, la usabilidad y la mantenibilidad. En el nivel 5, las organizaciones emplean ejemplos estadísticos, medidas del nivel y objetivos de fiabilidad para guiar el proceso de pruebas. El grupo de pruebas y el grupo de aseguramiento de la calidad software (SQA) trabajaran con los diseñadores y programadores para incorporar técnicas y herramientas que reduzcan los defectos y

mejoren la calidad del software. También se tiene en cuenta el coste de conseguir los objetivos de calidad y el coste de no hacerlo.

Subobjetivos.

- **SO1.** El grupo de pruebas y el grupo SQA establecerán los objetivos de calidad para los productos software.
- **SO2.** Los directores de pruebas incorporarán los objetivos definidos previamente en los planes de pruebas.
- **SO3.** Formar al grupo de pruebas en métodos estadísticos.
- **SO4.** Utilizar las entradas del usuario en el modelado.

Optimización del proceso de pruebas.

El proceso de pruebas es objeto de continuas mejoras en todos los proyectos de la organización en niveles altos de TMM. El proceso de prueba se cuantifica y se puede ser ajustar de manera que el crecimiento de la capacidad sea un proceso continuo. Existe una infraestructura en la organización para soportar la mejora continua, la cual consiste en un conjunto de políticas, estándares, formación, facilidades, herramientas y estructuras organizacionales.

Subobjetivos.

- **SO1.** Se establece el grupo de mejora del proceso de pruebas para seguir el proceso e identificar áreas de mejora.
- **SO2.** Implantar un mecanismo para evaluar nuevas herramientas y tecnologías que puedan mejorar la capacidad y madurez del proceso de pruebas.
- **SO3.** La efectividad del proceso de pruebas se debe evaluar de forma continua, y las decisiones cuando se paren las pruebas deben estar relacionadas con los objetivos de calidad y realizadas de una forma óptima y medible.

3.2.3 TMMI: Modelo de madurez de las pruebas integrado

Testing maturity model integrated (TMMI) [GOSLIN08] es la evolución de TMM, desarrollada por “TMMI foundation” como guía y marco de referencia para la mejora del proceso de pruebas. Dicho modelo surgió como respuesta y complemento de CMMI, ya que su enfoque, al ser más amplio y estructurado, logra un impacto favorable sobre la calidad del producto.

Su estructura se desarrolla en consonancia con la de CMMI utilizando áreas de proceso, objetivos genéricos, prácticas genéricas, objetivos y prácticas específicas. Aunque el modelo es bastante fácil de entender, su aplicación en una organización no siempre es una tarea simple.

3.2.3.1 Estructura del modelo

La estructura de TMMi se basa principalmente en la estructura de CMMI, presentando una mejora en los procesos relativos a los procesos de prueba y definiendo también niveles de madurez [GOSLIN08]. Aunque a diferencia de CMMi, hasta el momento solo se ha desarrollado una representación por etapas, tal y como queda reflejada en la Versión 2.0 de TMMi publicada por TMMifoundation [GOSLIN08].

Los **niveles de madurez**, o la escala jerárquica evolutiva, de TMMi se encuentran divididos en un conjunto de **áreas de proceso**. Estas últimas se tratan de un conjunto de actividades que una organización necesita realizar para alcanzar un determinado nivel de madurez. Al igual que en CMMI, un área de proceso se consigue gracias al logro de una serie de prácticas claramente diferenciadas, clasificadas como componentes requeridos y esperados. A continuación en la Figura 11 se representa la estructura de TMMi y sus principales componentes:

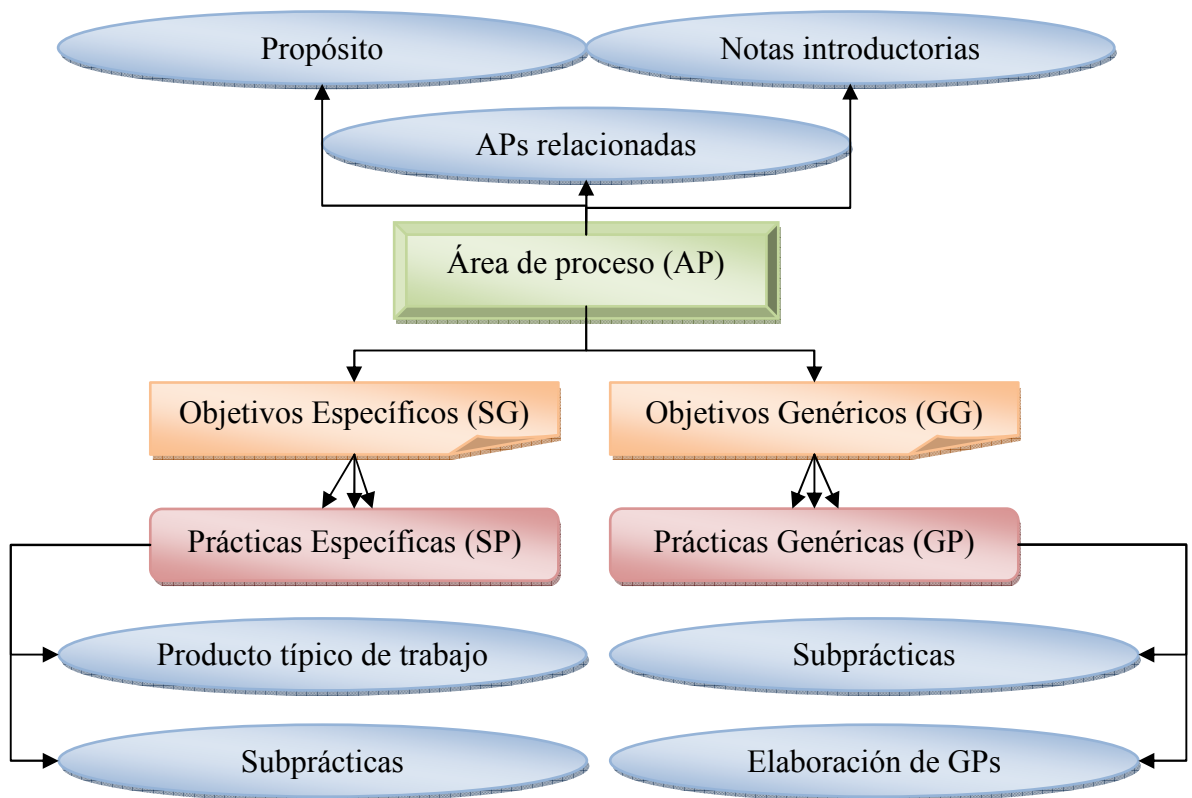


Figura 11. Estructura de TMMi

Los distintos componentes que forman TMMi se agrupan en tres categorías: los componentes informativos, los requeridos y los genéricos.

Los “*componentes informativos*” son distintas informaciones que sirven apoyo a la organización para afrontar otros componentes. Se conocen como componentes informativos de TMMi a: las **subprácticas**, que son descripciones detalladas para interpretación de una práctica específica; a los **productos típicos del trabajo**, que forman un salidas de una práctica específica; a las **notas introductorias**, las cuales describen los

principales conceptos incluidos en un área de proceso; a los **propósitos**, los cuales describen la funcionalidad de un área de proceso; al **alcance**, que identifica las prácticas de prueba que se abordan en dicho área de proceso; y a las **elaboraciones de las prácticas genéricas**, las cuales proveen de una guía de cómo una práctica genérica se debe realizarse.

Los “*componentes esperados*” describen qué debe implementar una organización para alcanzar un componente requerido. Se conocen como componentes esperados a: las **prácticas específicas**, que son descripciones de una actividad importante para conseguir un objetivo específico; y a las **prácticas genéricas**, las cuales describen actividades importantes para conseguir un objetivo genérico.

Los “*componentes requeridos*” describen qué debe alcanzar una organización para satisfacer un área de proceso. Se distinguen los **objetivos específicos** que describen características necesarias para satisfacer un determinado área de proceso; y los **objetivos genéricos** que describen características comunes a distintas áreas de proceso.

Representación

TMMi establece 5 **niveles de madurez** que permiten una evolución del proceso de pruebas desde un proceso caótico hasta un proceso gestionado, medido, definido y optimizado, facilitando su implantación con CMMI. La estructura interna es rica en prácticas de pruebas, pudiendo ser aplicadas en procesos de calidad.

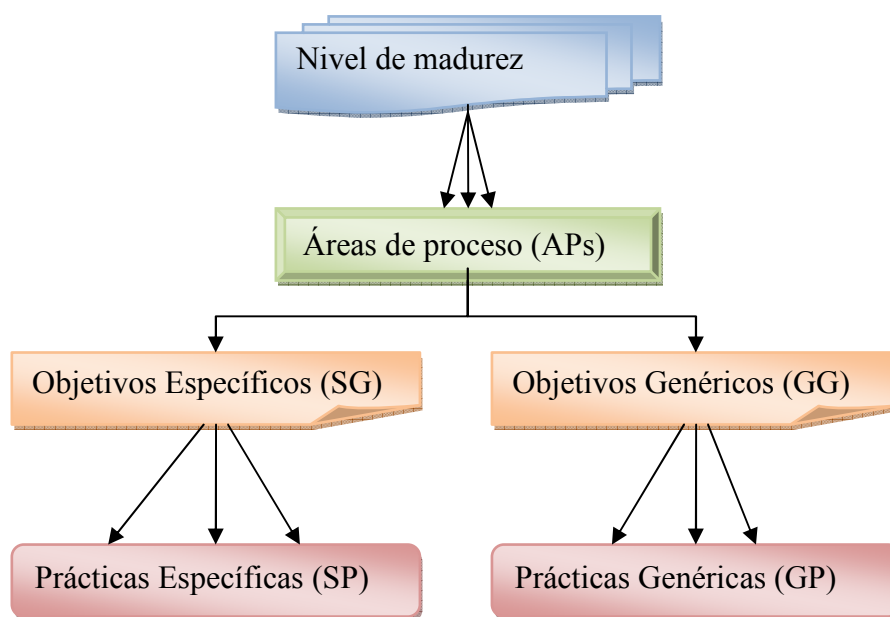


Figura 12. Estructura de la representación por etapas de TMMi.

Los niveles de madurez que una organización desea obtener determinan qué objetivos y prácticas genéricas deben aplicarse. Cuando se consigue alcanzar un nivel de madurez 2, se aplican las áreas de proceso del mismo y el objetivo genérico 2 y sus respectivas prácticas. Sin embargo, cuando se trata de implementar el objetivo genérico 3, o uno superior, es necesario haber completado y mejorar los niveles anteriores. Esto se

aplica a todas las áreas de proceso menos a la *Política y estrategia de pruebas*. Cabe destacar que, en la actualidad, aunque el modelo si se encuentra definido en su totalidad, sólo se encuentra desarrollado hasta el nivel 3 [GOSLIN08].

Nivel 1-Inicial: En el primer nivel de madurez, el proceso de pruebas es caótico e indefinido y a menudo es considerado una parte de la depuración de errores [GOSLIN08]. Esto se debe en gran parte a que la organización no proporciona un ambiente estable para dar soporte al proceso de pruebas. Dependiendo el éxito de las mismas de la capacidad y de la valía de los empleados y no del uso procesos establecidos. La ausencia de planificación en las pruebas provoca, que se desarrollen de forma desorganizada y una vez completada la codificación. Además, se produce una carencia de recursos, de herramientas y de personal con la formación necesaria. Por todo esto, el objetivo del proceso de pruebas en este nivel se limita a demostrar que el software se ejecuta sin errores importantes. Resultando productos que son lanzados sin la visibilidad adecuada en relación con la calidad y los riesgos.

En este nivel, las organizaciones tienden a abandonar el proceso de pruebas en tiempos de crisis y suelen ser incapaces de repetir sus logros. Asimismo, suelen excederse en el tiempo de entrega de sus productos y la calidad no suele ser acorde con las expectativas.

Nivel 2-Gestionado: En el segundo nivel de madurez, el proceso de pruebas se convierte en un proceso gestionado y se establece una clara distinción con la depuración de errores [VEENENDAAL06]. Esto facilita que las prácticas existentes en este nivel se conserven durante épocas de crisis. Sin embargo, las pruebas siguen siendo consideradas por muchos de los implicados en el proyecto como una tarea externa, que comienza una vez terminada la codificación.

El principal objetivo en este nivel es el de verificar que el producto satisface los requerimientos marcados. Para llegar a este objetivo y para promover la mejora del proceso, se establecen guías que marquen el camino a seguir dentro del proyecto y se desarrollan planes de pruebas. Estos planes, definen lo que se requiere probar, cuando, cómo y por quién. Se encuentran basados en la evaluación del riesgo del producto, gracias a la utilización de técnicas de gerencia de riesgos, que identifican los riesgos del producto basándose en los requisitos documentados.

Posteriormente, se realiza el seguimiento y el control de las pruebas para asegurar que se desarrollan según lo planeado. Se atribuye la responsabilidad del control del estado de los productos y de la entrega de servicios a los gerentes. Para derivar y seleccionar casos de pruebas se aplican técnicas de diseño de pruebas de las especificaciones. A pesar de las medidas tomadas en este nivel, se producen muchos problemas al llevarse a cabo el proceso en fases tardías del ciclo de vida del desarrollo software. Esto da lugar a que los defectos se propaguen desde los requerimientos y el diseño al código y todavía no existen programas de revisiones formales para hacer frente a ésta cuestión [GOSLIN08].

Nivel 3-Definido: En el tercer nivel de madurez, las pruebas ya no son una fase más posterior a la codificación en una organización. Se integran completamente en ciclo de vida del desarrollo. La estructura que toma plan de pruebas en éste nivel se desarrolla a partir de las características adquiridas en la planificación del nivel 2. Pero, al estar los

procesos descritos con mayor rigurosidad se deben revisar las áreas de proceso del nivel de madurez 2. Además, se debe adaptar el conjunto de procesos estándar de la organización a un determinado proyecto, para obtener una mayor consistencia en las pruebas, a excepción de las diferencias permitidas por las propias directrices establecidas.

Las organizaciones en este nivel comienzan a darse cuenta de la importancia de las revisiones en el control de la calidad, realizando para ello un programa formal de revisiones, aunque estas todavía no se encuentran vinculadas al proceso de pruebas dinámicas. Las revisiones se suceden durante todo el ciclo de vida, implicándose los probadores en las revisiones de las especificaciones de requisitos. Además, para reforzar el trabajo realizado en las pruebas, se reúnen, se almacenan y se gestionan los casos de prueba en una base de datos central. Allí, se realiza la reutilización, consulta y revisión de las pruebas.

Nivel 4-Gerencia y medida: En el cuarto nivel de madurez, la organización de las pruebas se encuentra totalmente formada, definiendo correctamente un proceso medible. Por ello, la organización y los proyectos establecen objetivos cuantitativos como: la fiabilidad, la utilidad y la capacidad de mantenimiento. Estos objetivos tienen la finalidad de obtener la calidad requerida del producto, utilizándolos como criterios de calidad en los procesos durante su desarrollo [BLACK], y dando lugar al estudio, en términos estadísticos, de la calidad del producto y del rendimiento de los procesos durante todo el ciclo de vida.

Tanto las revisiones como las inspecciones son consideradas en este nivel como una parte más de las pruebas, y son utilizadas como una medida útil para la elaboración del informe de calidad. Percibiéndose finalmente las pruebas como una evaluación formada por todas las actividades del ciclo de vida que se encuentran relacionadas con el control de los productos. Una vez realizadas las medidas necesarias, estas se almacenan para dar lugar a una base de hechos útil con el fin de afrontar futuras tomas de decisiones.

Nivel 5-Optimización: En el quinto nivel de madurez, debido a la infraestructura proporcionada por el logro de los objetivos en los niveles anteriores, se define en su totalidad el proceso de pruebas, siendo posible controlar los costes y la eficacia del mismo [GOSLIN08]. No por esto, la organización debe descuidar la mejora continua de sus procesos, basándose para ello en el estudio cuantitativo de las causas comunes de la variación inherente al proceso.

Para mejorar el funcionamiento del proceso de prueba, se realizan mejoras tecnológicas y del propio proceso de forma incremental. Además, con tal de fomentar la mejora continua, tanto los métodos y como las técnicas optimizadas son objetos de una atención constante. En este nivel cobra importancia el hecho de que se practique el control de calidad y la prevención de defectos, llegando a ser considerados como áreas de proceso. Asimismo, se promueve el uso de medidas de calidad basadas en el muestreo estadístico y en la medida de los niveles de confianza, dotando al proceso de fiabilidad y confianza.

NIVEL DE MADUREZ	Objetivos de madurez
OPTIMIZACIÓN	Prevención de los defectos. Optimización de los procesos de prueba. Control de la calidad.

NIVEL DE MADUREZ	Objetivos de madurez
GESTION MEDIDA Y	Medición de la prueba. Evaluación de la calidad del producto. Revisiones por pares avanzadas.
DEFINIDO	Organización de las pruebas. Programa de formación de las pruebas. Integración y ciclo de vida de las pruebas. Pruebas no funcionales. Revisiones por pares.
GESTIONADO	Estrategia y política de pruebas. Planificación de pruebas. Control y supervisión de pruebas. Diseños y ejecución de pruebas. Entorno de pruebas.
INICIAL	No tiene.

Tabla 6. Tabla de relación de niveles de madurez y APs en TMMi.

3.2.3.2 Áreas de proceso

Las **áreas de proceso (AP)** [GOSLIN08] indican las cuestiones que deben abordarse para conseguir un nivel de madurez. Cada AP identifica un conjunto de actividades de prueba relacionadas. Cuando se realizan todas sus prácticas, se produce una considerable mejora en las actividades relacionadas con el área de proceso y ésta se considera realizada. En TMMi, sólo aquellas áreas de proceso que son identificadas, son consideradas como factores determinantes de la capacidad del proceso de pruebas.

Aunque TMMi puede ser utilizado en solitario, al poder ser utilizado como complemento de CMMi, en muchos casos un nivel de TMMi necesita el apoyo específico de las áreas de proceso correspondientes a su respectivo nivel de CMMi o a uno superior. Normalmente las APs y las prácticas que ya están incluidas en CMMi sólo son referenciadas en TMMi. En la actualidad sólo se encuentran desarrolladas las áreas de proceso específicas del nivel 2 y 3 [GOSLIN08].

Al igual que pasa con CMMI, las áreas de proceso cuanta con la ayuda de distintos componentes que favorecen su desarrollo, así en TMMi se destacan los componentes informativos, los componentes requeridos y los componentes esperados.

Los **componentes informativos** proporcionan detalles que ayudan a las organizaciones a comenzar a pensar en cómo aproximarse a los componentes requeridos y esperados. A continuación, se distinguen los distintos componentes informativos presentes en TMMi [GOSLIN08].

- Las *subprácticas*, que se tratan de descripciones detalladas que proveen de una guía para la interpretación e implementación de una práctica específica.
- El *producto típico del trabajo*, los cuales ofrecen una lista de ejemplos de salida de una práctica específica.

- El *propósito*, que describe el objetivo del área de proceso.

Los **componentes requeridos** describen lo que una organización debe realizar para satisfacer un área de proceso. Estos logros, se deben implementar de forma visible en los procesos de una organización. Utilizándose la satisfacción de los objetivos en las evaluaciones, como base para determinar si un área de proceso ha sido realizada y satisfecha. Los componentes requeridos en TMMI [GOSLIN08] son los *objetivos genéricos* y los *objetivos específicos*.

- Un *objetivo genérico (GO)* [GOSLIN08] surge cuando está próximo el final de un área de proceso y son llamados genéricos a causa de que aparecen en distintas áreas de proceso. Un objetivo genérico describe las características que deben de presentar las áreas de proceso para ser institucionalizadas. Son utilizados para determinar cuándo se satisfacen las necesidades de un área de proceso.
- Un *objetivo específico (SG)* describe características únicas que deben estar presentes para satisfacer un área de proceso. Son utilizados en las evaluaciones para ayudar a determinar si se cumple un área de proceso [GOSLIN08].

Los **componentes esperados** describen lo que una organización puede implementar para lograr un componente requerido, sirviendo de guía a quien implementa las mejoras o las evaluaciones. Estos componentes incluyen tanto prácticas específicas como prácticas genéricas, las cuales deberán estar presentes en los procesos planificados e implementados para que los objetivos puedan considerarse satisfechos [GOSLIN08].

- Las *prácticas genéricas (GP)* aparecen cerca del final de área de proceso y son llamadas genéricas al aparecer en distintas áreas de proceso. Una GP es la descripción de una actividad que es considerada importante para conseguir un objetivo genérico asociado [GOSLIN08]. Las prácticas genéricas describen aspectos importantes para la institucionalización de procesos.
- Una *práctica específica (SP)* [GOSLIN08] es una descripción de una actividad que es considerada importante para conseguir su objetivo específico asociado. Las SP describen las actividades esperadas para conseguir un objetivo específico de un determinado área.

Los objetivos y las prácticas genéricas se derivan en gran medida de CMMI, a excepción de los objetivos pertenecientes a la representación continua. Esto es debido a que de momento no tiene existe la representación continua en TMMI. También es adoptada la numeración de CMMI para estos elementos con el fin de evitar confusiones. Por todo esto, sólo aparecen reflejados en TMMI GG2, GG3 y sus respectivas prácticas. A continuación en la Tabla 7. se muestra la relación entre los distintos niveles de TMMI y sus respectivos objetivos globales.

Nivel	GG1	GG2	GG3	GG4	GG5
Nivel de madurez 2		X			
Nivel de madurez 3			X		
Nivel de madurez 4			X		
Nivel de madurez 5			X		

Tabla 7. Relación entre niveles de madurez y objetivos genéricos.

Por todo ello, no se repiten ni los objetivos ni las prácticas genéricas que aparecen en CMMI. Simplemente se referencia, pudiéndose comprobar tanto GG2 y GG3 en la sección 3.2.1.2 del capítulo de CMMI que aparece en este mismo texto. Así mismo, también aparecen reflejadas las correspondientes prácticas de los objetivos genéricos en las secciones anteriormente mencionadas.

A continuación se detallan las distintas áreas de proceso reflejadas en TMMI y la relación existente con sus respectivos objetivos y prácticas específicas:

Política y estratégica de pruebas.

En éste área de proceso se trata de desarrollar y establecer una política y una estrategia donde se establezca cierto nivel de pruebas, para poder medirlo se utilizan medidores de realización de las pruebas. Se desarrolla bajo el soporte del GG2.

Objetivos y prácticas específicas.

- **SG1.** Establecer una política de pruebas.
 - **SP1.1.** Definir los objetivos del plan de pruebas.
 - **SP1.2.** Definir la política de pruebas.
 - **SP1.3.** Distribuir la política de pruebas a las partes interesadas.
- **SG2.** Establecer una estrategia de pruebas.
 - **SP2.1.** Definir la estrategia de pruebas.
 - **SP2.2.** Distribuir la estrategia de pruebas a las partes interesadas.
- **SG3.** Establecer indicadores de la realización de las pruebas.
 - **SP3.1.** Definir indicadores de la realización de las pruebas.
 - **SP3.2.** Desarrollar los indicadores de la realización de las pruebas.

Plan de pruebas.

El propósito del plan de pruebas consiste en definir una aproximación a las pruebas basadas en la identificación de riesgos y en la definición de una estrategia. Para ello, establece y mantiene el buen funcionamiento de la planificación, con el fin de realizar y mejorar las actividades de pruebas. Se desarrolla bajo el soporte del GG2 y del GG3.

Objetivos y prácticas específicas.

- **SG1.** Asegurar los riesgos del producto.
 - **SP1.1.** Definir la categoría y los parámetros de los riesgos del producto.
 - **SP1.2.** Identificar los riesgos del producto.
 - **SP1.3.** Analizar los riesgos del producto.
- **SG2.** Establecer una aproximación a las pruebas.
 - **SP2.1.** Identificar los temas y las características para probar.
 - **SP2.2.** Definir la aproximación de pruebas.
 - **SP2.3.** Definir criterios de entrada.
 - **SP2.4.** Definir los criterios de salida.
 - **SP2.5.** Definir criterios de suspensión y reanudación.
- **SG3.** Estimar las pruebas.
 - **SP3.1.** Establecer una jerarquía de trabajo de alto nivel.
 - **SP3.2.** Definir el ciclo de vida de las pruebas.
 - **SP3.3.** Estimar esfuerzo y coste de las pruebas.
- **SG4.** Desarrollar el plan de pruebas.
 - **SP4.1.** Establecer el calendario de pruebas,
 - **SP4.2.** Planificar el personal de pruebas.
 - **SP4.3.** Plan para involucrar a las partes interesadas.
 - **SP4.4.** Identificar los riesgos del plan de proyecto.
 - **SP4.5.** Establecer el plan de pruebas.
- **SG5.** Obtener compromiso con el plan de pruebas.
 - **SP5.1.** Revisar el plan de pruebas.
 - **SP5.2.** Equiparar carga de trabajo con los recursos disponibles.
 - **SP5.3.** Obtener compromiso con el plan de pruebas.

Seguimiento y control de las pruebas.

El propósito de este AP es proveer y comprender el progreso de las pruebas y la calidad del producto para poder adoptar acciones correctivas cuando sea necesario. Se desarrolla bajo el soporte del GG2 y GG3.

Objetivos y prácticas específicas.

- **SG1.** Seguir el progreso de las pruebas en el plan.
 - **SP1.1.** Seguir los parámetros del plan de pruebas.
 - **SP1.2.** Seguimiento de los recursos del entorno de pruebas utilizados.
 - **SP1.3.** Seguimiento del compromiso de las pruebas.
 - **SP1.4.** Seguimiento de los riesgos del proyecto de pruebas.
 - **SP1.5.** Seguimiento de la involucración de las partes interesadas.
 - **SP1.6.** Estudiar las pruebas de calidad del producto.
 - **SP1.7.** Estudiar el progreso de los hitos del producto.

- **SG2.** Seguimiento de la calidad del producto a través de lo planificado.
 - **SP2.1.** Confirmar criterios de entrada.
 - **SP2.2.** Seguimiento de los incidentes de las pruebas.
 - **SP2.3.** Seguimiento de los riesgos del producto.
 - **SP2.4.** Seguimiento de los criterios de salida.
 - **SP2.5.** Seguimiento de los criterios de suspensión
 - **SP2.4.** Realizar estudios de la calidad del producto.
 - **SP2.5.** Realizar estudios de los hitos de calidad del producto.
- **SG3.** Gestionar acciones correctivas para el cierre.
 - **SP3.1.** Analizar las cuestiones.
 - **SP3.2.** Tomar acciones correctivas.
 - **SP3.3.** Gestionar acciones correctivas.

Diseño y ejecución de las pruebas.

El propósito del diseño y ejecución de las pruebas consiste en mejorar la capacidad del proceso de pruebas durante la fase de diseño y ejecución. Para ello, establece especificaciones y técnicas de diseño, realizando una ejecución estructurada del proceso de pruebas y gestionando los incidentes de las pruebas para su clausura. GG2 y 3.

Objetivos y prácticas específicas.

- **SG1.** Realización de análisis y diseño de pruebas a través de técnicas de pruebas.
 - **SP1.1.** Identificar y dar prioridades a las condiciones de pruebas.
 - **SP1.2.** Identificar y dar prioridades a los casos de pruebas.
 - **SP1.3.** Identificar datos de pruebas específicos necesarios.
 - **SP1.4.** Mantener el seguimiento horizontal con los requisitos.
- **SG2.** Realizar la implementación de las pruebas.
 - **SP2.1.** Desarrollar y priorizar los procedimientos de pruebas.
 - **SP2.2.** Crear datos específicos de pruebas.
 - **SP2.3.** Especificar el proceso de pruebas de admisión.
 - **SP2.4.** Desarrollar el calendario de ejecución de pruebas.
- **SG3.** Realizar ejecución de pruebas.
 - **SP3.1.** Realizar pruebas de entrada.
 - **SP3.2.** Ejecutar casos de uso.
 - **SP3.3.** Reportar los incidentes de las pruebas.
 - **SP3.4.** Escribir el log de pruebas.
- **SG4.** Gestionar incidentes de las pruebas para la clausura.
 - **SP4.1.** Decidir los incidentes de pruebas en la configuración.
 - **SP4.2.** Realizar acciones apropiadas para solucionar incidentes.

Entorno de pruebas

El propósito del entorno de las pruebas es establecer y mantener un entorno adecuado, incluyendo los datos de pruebas, donde sea posible ejecutar las pruebas de una forma gestionada y repetitiva. Se desarrolla bajo el soporte del GG2 y del GG3.

Objetivos y prácticas específicas.

- **SG1.** Desarrollar los requerimientos del entorno de pruebas.
 - **SP1.1.** Obtener las necesidades del entorno de pruebas.
 - **SP1.2.** Desarrollar los requerimientos del entorno de pruebas.
 - **SP1.3.** Analizar los requerimientos del entorno de pruebas.
- **SG2.** Desarrollar la implementación del entorno de pruebas.
 - **SP2.1.** Implementar el entorno de pruebas.
 - **SP2.2.** Crear datos de prueba genéricos.
 - **SP2.3.** Especificar procedimientos de prueba del entorno de prueba de entrada.
 - **SP2.4.** Realizar pruebas de entrada del entorno de pruebas.
- **SG3.** Gestionar y controlar los entornos de pruebas.
 - **SP3.1.** Realizar administración de sistemas.
 - **SP3.2.** Realizar administración de datos de pruebas.
 - **SP3.3.** Coordinar la disponibilidad y el uso de los entornos de prueba.
 - **SP3.4.** Reportar y gestionar los incidentes de los entornos de prueba.

Organización de las pruebas.

El propósito de la organización de las pruebas consiste en identificar e organizar un grupo de personas altamente cualificadas, que van a ser responsables de las pruebas. Este grupo puede gestionar la mejora del proceso de pruebas de la organización y de los activos del proceso de pruebas basándose en el profundo conocimiento de las fortalezas y debilidades de los mismos. Se desarrolla bajo el soporte del GG3.

Objetivos y prácticas específicas.

- **SG1.** Establecer una organización de pruebas.
 - **SP1.1.** Definir la organización de las pruebas.
 - **SP1.2.** Obtener compromisos de la organización de las pruebas.
 - **SP1.3.** Implementar la organización de las pruebas.
- **SG2.** Establecer funciones de las pruebas para los especialistas.
 - **SP2.1.** Identificar las funciones de las pruebas.
 - **SP2.2.** Desarrollar descripciones de los trabajos.
 - **SP2.3.** Asignar personal a las funciones de las pruebas.
- **SG3.** Establecer trayectorias de carreras de las pruebas.
 - **SP3.1.** Establecer trayectorias de las carreras de las pruebas.

- **SP3.2.** Desarrollar planes personales de carreras de pruebas.
- **SG4.** Determinar, planear e implementar las mejoras del proceso de pruebas.
 - **SP4.1.** Evaluar los procesos de pruebas de la organización.
 - **SP4.2.** Identificar las mejoras de los procesos de pruebas de la organización.
 - **SP4.3.** Planificar las mejoras del proceso de pruebas.
 - **SP4.4.** Implementar las mejoras del proceso de pruebas.
- **SG5.** Desarrollar la organización de los procesos de prueba e incorporar las lecciones aprendidas.
 - **SP5.1.** Desarrollar los procesos de pruebas estándar y sus principios.
 - **SP5.2.** Seguir la implementación.
 - **SP5.3.** Incorporar las lecciones aprendidas en los procesos de pruebas organizativos.

Programa de formación de las pruebas.

El propósito del programa de formación de las pruebas es el de desarrollar un programa de formación que facilite el desarrollo de los conocimientos y de las características del personal que va a realizar actividades de pruebas. Esto se realiza con el fin de que los distintos roles se puedan llevar a cabo de la manera más efectiva y eficiente.

Objetivos y prácticas específicas.

- **SG1.** Establecer la capacidad de formar una organización de pruebas.
 - **SP1.1.** Establecer las estrategias de las necesidades de formación de las pruebas.
 - **SP1.2.** Alinear las necesidades organizativas y de formación de las pruebas del proyecto.
 - **SP1.2.** Establecer un plan de formación de pruebas organizacional.
 - **SP1.4.** Establecer la capacidad de formación de las pruebas.
- **SG2.** Facilitar la formación de pruebas necesarias.
 - **SP2.1.** Proporcionar formación de pruebas.
 - **SP2.2.** Establecer registros de la formación de las pruebas.
 - **SP2.3.** Evaluar la eficacia de la formación de las pruebas.

Integración y ciclo de vida de las pruebas.

El propósito de la integración y el ciclo de vida de las pruebas es establecer y mantener un conjunto aprovechable de principios del proceso de pruebas organizacional y de estándares del entorno de trabajo e integrarlos, además de sincronizar el ciclo de vida de las pruebas con el del desarrollo. También, define un enfoque coherente en los diferentes niveles de las pruebas, basándose en los riesgos identificados y en la estrategia de prueba definida. Finalmente presenta un plan de pruebas general basado en el ciclo de pruebas definido. Se desarrolla bajo el soporte del GG3.

Objetivos y prácticas específicas.

- **SG1.** Establecer los activos del proceso de pruebas organizacional.
 - **SP1.1.** Establecer los procesos de pruebas estándar.
 - **SP1.2.** Establecer descripciones del modelo del ciclo de vida de las pruebas en todos los niveles.
 - **SP1.3.** Establecer criterios y directrices de adaptación.
 - **SP1.4.** Establecer la base de datos de los procesos de prueba de la organización.
 - **SP1.5.** Establecer la colección de activos del proceso de prueba de la organización.
 - **SP1.6.** Establecer estándares de entornos de trabajo.
- **SG2.** Integrar los modelos del ciclo de vida con los modelos de desarrollo.
 - **SP2.1.** Establecer modelos del ciclo de vida integrados.
 - **SP2.2.** Revisar modelos del ciclo de vida integrados.
 - **SP2.3.** Obtener compromisos con el rol de las pruebas dentro de los modelos del ciclo de vida integrados.
- **SG3.** Establecer un plan de pruebas principal.
 - **SP3.1.** Realizar la evaluación del riesgo del producto.
 - **SP3.2.** Establecer el enfoque de las pruebas.
 - **SP3.3.** Establecer estimaciones de pruebas.
 - **SP3.4.** Definir la organización.
 - **SP3.5.** Desarrollar el plan de pruebas principal.
 - **SP3.6.** Obtener compromiso con el plan de pruebas principal.

Pruebas no funcionales.

El propósito de las pruebas no funcionales es el de mejorar la capacidad del proceso de pruebas para pruebas no funcionales durante la planificación, diseño y ejecución de las mismas. Esto se realiza mediante la definición de un enfoque de las pruebas basado en la identificación de los riesgos del producto, las especificaciones y la ejecución de un proceso de pruebas no funcionales.

Objetivos y prácticas específicas.

- **SG1.** Realizar el aseguramiento de los riesgos del producto no funcionales.
 - **SP1.1.** Identificar los riesgos del producto no funcionales.
 - **SP1.2.** Analizar los riesgos del producto no funcionales.
- **SG2.** Establecer un enfoque a las pruebas no funcionales.
 - **SP2.1.** Identificar las características que deben de ser probadas.
 - **SP2.2.** Define el enfoque a las pruebas no funcionales.
 - **SP2.3.** Define el criterio de salida no funcional.

- **SG3.** Diseñar y analizar el análisis de las pruebas no funcionales.
 - **SP3.1.** Identificar y establecer prioridades de las condiciones de las pruebas no funcionales.
 - **SP3.2.** Identificar y establecer prioridades de los casos de prueba no funcionales.
 - **SP3.3.** Identificar los datos de prueba específicos necesarios.
 - **SP3.4.** Mantener un seguimiento horizontal con los requerimientos no funcionales.
- **SG4.** Implementar pruebas no funcionales.
 - **SP4.1.** Desarrollar y establecer prioridades de los procedimientos de pruebas no funcionales.
 - **SP4.2.** Crear datos de prueba específicos.
- **SG5.** Ejecutar las pruebas no funcionales.
 - **SP5.1.** Ejecutar casos de prueba no funcionales.
 - **SP5.2.** Informar de los incidentes en las pruebas no funcionales.
 - **SP5.4.** Escribir logs de las pruebas.

Aunque puede llegar a utilizarse de forma aislada, TMMi tiene un gran peso como complemento de CMMi. Como resultado, en muchos casos un nivel de TMMi necesita del soporte específico de las áreas de proceso de CMMi de su mismo nivel o de las superiores. A continuación, se detalla en qué partes de los niveles de TMMi ofrece soporte CMMi con sus áreas de proceso:

APs nivel 2 CMMI	Nivel 2 TMMI
Gestión de la configuración (CM).	GP2.6. Planificación del proceso.
Aseguramiento de la calidad de proceso y de producto (PPQA).	GP2.9. Evaluar objetivamente su aplicación.
Supervisión y control del proyecto (SAM).	AP-Control y seguimiento del proyecto
Planificación del proyecto (PP).	AP-Planificación de las pruebas. GP2.7. Identificar e involucrar las partes interesadas importantes.
Medición y análisis (MA).	AP-Estrategia y política de pruebas.
Gestión de requerimientos (REQM).	AP- Diseño y ejecución de las pruebas.

Tabla 8. Soporte del nivel 2 de CMMI en el nivel 2 de TMMI.

APs nivel 3 CMMI	Nivel 2 TMMI
Gestión de riesgos (RSKM).	AP- Entorno de pruebas.
Desarrollo de requerimientos (RD).	AP-Planificación

Tabla 9. Soporte del nivel 3 de CMMI en el nivel 2 de TMMI.

APs nivel 2 CMMI	Nivel 3 TMMI
Gestión de la configuración (CM).	GP2.6. Planificación del proceso.
Aseguramiento de la calidad de proceso y de producto (PPQA).	GP2.9. Evaluar objetivamente su aplicación.

APs nivel 2 CMMI	Nivel 3 TMMI
Planificación del proyecto (PP).	AP- Integración y ciclo de vida de las pruebas.

Tabla 10. Soporte del nivel 2 de CMMI en el nivel 3 de TMMI.

APs nivel 3 CMMI	Nivel 3 TMMI
Enfoque en procesos de la organización (OPF).	AP-Organización de las pruebas.
Definición de proceso de la organización (OPD).	AP- Integración y ciclo de vida de las pruebas.
Formación organizativa (OT).	AP-Programa de formación de pruebas.
Verificación (VER).	AP-Revisiones por pares.

Tabla 11. Soporte del nivel 3 de CMMI en el nivel 3 de TMMI.

3.2.4 TPI: Modelo de mejora del proceso de pruebas

El grupo SOGETI desarrolló el *modelo de mejora del proceso de pruebas (TPI - Test Process Improvement)* [KOOMEN99], el cual se desarrolla a través del conocimiento y la experiencia de las pruebas de control del software. Éste es un medio de ayuda para mejorar el proceso de pruebas, permitiendo establecer el nivel de madurez del proceso de pruebas dentro de una organización [FREIDKES07]. Partiendo de este criterio, el modelo ayuda a definir pasos de mejora graduales y controlados [SCHIJNDEL]. TPI se basa en las mejores prácticas de la industria relativas a la mejora del proceso de pruebas [DUMKE]. Incluye guías prácticas para evaluar el nivel de madurez de las pruebas de una organización, así como de los pasos para mejorar el proceso de pruebas. Además, ofrece un marco para determinar las áreas débiles y fuertes del proceso. Este modelo se puede complementar con la metodología del proceso de pruebas (TMap), elaborada por el mismo grupo, que dota de una mayor rapidez a la elaboración de pruebas.

3.2.4.1 Estructura del modelo

TPI tiene en cuenta los diversos aspectos del proceso de pruebas, tales como el uso de herramientas de pruebas, las técnicas de diseño de pruebas o la realización de informes. En su análisis, se hacen visibles los puntos fuertes y débiles del proceso de pruebas. Estos aspectos se denominan *áreas clave*. El modelo TPI se estructura de la siguiente manera:

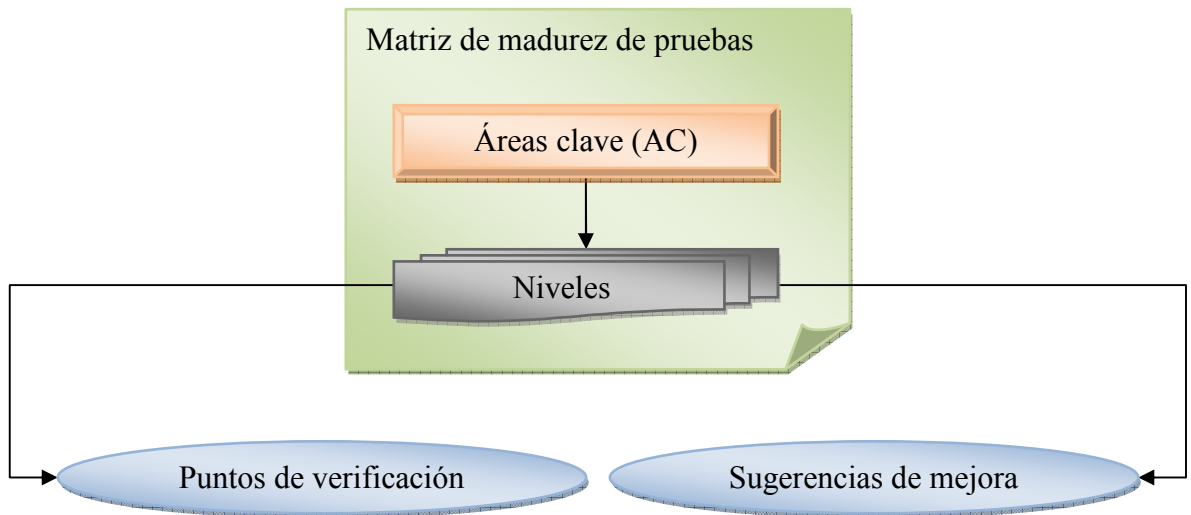


Figura 13. Estructura de TPI.

TPI [KOOMEN99] se compone de 20 áreas clave, cada una con diferentes **niveles de madurez**. Niveles utilizados para clasificar el estado de la A-C, identificando la madurez de cada una de ellas. Un nivel más alto implica más eficiencia desde el punto de vista de tiempo, de coste y/o de calidad. Todos los niveles y áreas clave están interrelacionados en la **matriz de madurez** (ver Figura 13) del proceso de pruebas [KOOMEN99]. Ésta proporciona una forma adecuada de definir las prioridades internas, teniendo en cuenta las dependencias entre las áreas clave y los niveles correspondientes. En la matriz, cada nivel está relacionado con una determinada escala de madurez del proceso de pruebas.

Las áreas clave, los niveles y la matriz se utilizan para identificar los puntos fuertes y débiles del proceso de pruebas, para definir el nivel de madurez que queremos alcanzar, y para establecer las acciones para llevar a cabo la mejora. El uso de la matriz permite seleccionar y evaluar propuestas de mejora con mayor facilidad.

Los requisitos de cada nivel están definidos en los **puntos de verificación**, que se deben cumplir para acceder al nivel correspondiente. Por tanto, los puntos de verificación proporcionan una mejora objetiva de clasificación. Por último, el modelo ofrece **sugerencias de mejora** que son consejos para alcanzar cierto nivel de madurez de las pruebas [KOOMEN99]. Tratan de optimizar el proceso de pruebas, pero a diferencia de los puntos de verificación, su uso no es necesario.

Representación

Después de determinar los niveles para cada área clave, se debe dirigir la atención hacia cuáles son los pasos de mejora que hay que realizar [SWINKELS]. Esto se debe a que no todas las áreas clave y sus niveles tienen la misma importancia. Además de estas prioridades, existe una dependencia entre los niveles de diferentes áreas clave. Por ejemplo, antes de poder recibir estadísticas de los defectos encontrados (nivel A del Área Clave “Métricas”), el proceso de pruebas tiene que calificar para el nivel B del Área Clave “Manejo de Defectos”. Por lo tanto, todos los Niveles y Áreas Clave están

interrelacionados en una *matriz de madurez de pruebas*. La cual, se ha concebido como una buena manera de expresar las prioridades internas y dependencias entre los niveles y las áreas clave. Resultando su propósito principal el de mostrar los puntos fuertes y débiles del actual proceso de pruebas y ofrecer una ayuda a la hora de determinar la prioridad de las acciones de mejora [KOOMEN99]. Por ello, ofrece a todos los participantes una visión clara de la situación actual del proceso de pruebas siempre que se empleen datos reales.

En la matriz, cada nivel está relacionado con cierta escala de madurez de pruebas. Resultando así 13 escalas de madurez de pruebas. Las celdas abiertas entre diferentes niveles no tienen significado por sí mismas, pero indican que el lograr una mayor madurez para un área clave está relacionado con la madurez de otras áreas [KOOMEN99]. De todas formas, no existe graduación entre niveles: mientras que un proceso de pruebas no esté clasificado enteramente como nivel B, permanecerá en el nivel A.

Área Clave/Escala (Niveles)	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Estrategia de pruebas		A					B				C		D	
Modelo del ciclo de vida		A			B									
Momento de involucración			A				B				C		D	
Estimación y planificación				A							B			
Técnicas de diseño de pruebas		A		B										
Técnicas de pruebas estáticas					A		B							
Métricas						A			B			C		D
Herramientas de prueba					A			B			C			
Entorno de pruebas				A				B						C
Entorno de oficina				A										
Compromiso y motivación		A				B						C		
Funciones de pruebas y capacitación				A			B			C				
Alcance de la metodología					A						B			C
Comunicación			A		B							C		
Informes		A			B		C					D		
Gestión de defectos		A				B		C						
Administración de los elementos de las pruebas			A			B				C				D
Administración del proceso de pruebas		A		B								C		
Revisión estructurada							A			B				
Pruebas de caja blanca					A		B		C					

Tabla 12. Matriz de madurez de pruebas de TPI.

Es obvio que cada área clave no tendrá la misma atención ni profundidad, ya que cada proceso de pruebas tiene sus puntos fuertes y débiles. Por ello, la forma en que se organizan las áreas clave dentro de un proceso de pruebas determina su madurez [SWINKELS]. Debido a esto, y con el fin de permitir una visión del estado de cada área clave, TPI proporciona cuatro *niveles* (A, B, C, D). En los cuales, si uno es mayor que

otro, siendo ($A < B < C < D$), se produce una mejora sobre el nivel previo en factores de tiempo, coste y calidad [ANDER04].

Así, cada nivel consiste en el cumplimiento de ciertos requisitos (o puntos de verificación) por parte del área clave. Dicho cumplimiento asimismo, compromete el de los niveles anteriores [SCHIJNDEL]. Sin embargo, un proceso de pruebas puede llegar a no satisfacer los requisitos del nivel A, considerado el nivel más bajo, y por tanto se considera que dicha área se encuentra en un nivel indefinido. A continuación, en el siguiente apartado se establecen los distintos niveles pertenecientes a cada área clave que se refleja en el modelo [KOOMEN99] [SCHIJNDEL].

3.2.4.2 Áreas clave

En cada proceso de pruebas existen ciertas áreas que necesitan una atención específica con el fin de lograr un proceso bien definido. Estas áreas se denominan en TPI **áreas clave (AC)**, las cuales constituyen la base para mejorar y estructurar el proceso de pruebas.

El conjunto de las veinte áreas claves existentes en TPI cubren la totalidad del proceso de pruebas [KARINSALO] [KOOMEN99] [SCHIJNDEL]:

AC1. Estrategia de pruebas.

En la estrategia de pruebas se determinan los defectos (de calidad) descubiertos al realizar las pruebas. Debiendo estar dirigida a encontrar los defectos más importantes con la mayor rapidez y el menor coste posible. Cuantas más pruebas y medidas de detección se utilicen, es posible definir una mejor estrategia.

Niveles.

- A. Estrategia de Pruebas para una sola prueba.
- B. Estrategia de Pruebas combinada para pruebas de caja negra.
- C. Estrategia de Pruebas combinada para pruebas de caja negra y pruebas de caja blanca o evaluación.
- D. Estrategia de Pruebas combinada para todas las pruebas y actividades de evaluación.

AC2. Modelo del ciclo de vida.

La importancia de contar con un modelo de ciclo de vida reside en tener un mejor control del proceso de pruebas, dado que las actividades pueden ser planeadas y controladas consistentemente.

Dentro del proceso de pruebas se distinguen algunas fases: planificación, preparación, diseño, ejecución y cierre. En cada fase se efectúan algunas actividades y para cada actividad se registran aspectos tales como: los objetivos, las entradas, el proceso, los conceptos, las dependencias, las técnicas y las herramientas, y las instalaciones y la documentación.

Niveles.

- A. Planificación, Diseño, Ejecución.
- B. Planificación, Preparación, Diseño, Ejecución, Cierre.

AC3. Momento de involucración.

El momento de ejecución de las pruebas se inicia normalmente al concluir el diseño del software, pero el proceso de pruebas debe iniciarse mucho antes. La involucración temprana de las pruebas en el ciclo de vida del desarrollo de sistemas ayuda a detectar los defectos con mayor antelación y/o con más facilidad, e incluso ayuda a prevenir defectos. Todo ello da como resultado, una mejor coordinación entre las pruebas además de reducir enormemente la ruta crítica de las pruebas.

Niveles.

- A. Al terminar la especificación.
- B. Al inicio de la especificación.
- C. Al iniciar la definición de requerimientos.
- D. Al inicio del Proyecto.

AC4. Estimación y planificación.

La estimación y la planificación son la base para ahorrar capacidad y para coordinar las actividades de prueba y del proyecto. Es necesario estimar y planificar para definir las actividades que deben realizarse en cada momento y los recursos que serán necesarios.

Niveles.

- A. Estimación y planificación elemental.
- B. Estimación y planificación fundada estadísticamente.

AC5. Técnicas de diseño de pruebas.

Una técnica de diseño de pruebas se define como “un modelo para derivar casos de prueba de la documentación”. El uso de estas técnicas incrementa la visibilidad de la calidad y la cobertura de pruebas y conduce a una mayor reusabilidad de las pruebas. En base a una estrategia de pruebas, se utilizan diferentes técnicas de diseño de pruebas para obtener la cobertura del código de las partes de software cuyo alcance fue acordado.

Niveles.

- A. Técnicas Informales.
- B. Técnicas Formales

AC6. Técnicas de pruebas estáticas.

Las pruebas estáticas son validaciones de los productos sin ejecutar los programas o la evaluación de métricas de calidad específicas. Son utilizadas puesto que no siempre se pueden utilizar pruebas dinámicas. Además, las listas de verificación y mecanismos similares son muy útiles en este caso.

Niveles.

- A. Pruebas en base a entradas.
- B. Listas de Verificación (Checklists).

AC7. Métricas.

Las métricas son observaciones cuantitativas (mediciones). Para el proceso de pruebas, medir el progreso y la calidad del software probado es muy importante, así como también lo son las métricas en estas áreas. Las métricas se utilizan para poder administrar el proceso de pruebas, para poder tener evidencia al momento de expresar una opinión, y también para comparar diferentes sistemas o procesos.

Niveles.

- A. Estadísticas del proyecto (producto).
- B. Estadísticas del proyecto (Proceso).
- C. Estadísticas del sistema.
- D. Estadísticas de la organización

AC8. Herramientas de prueba.

Las herramientas de pruebas facilitan la automatización del proceso y con ella se consigue alguna de las metas siguientes: menor consumo de recursos, menos consumo de tiempo, mejor cobertura de pruebas, mayor flexibilidad, mayor o más rápida comprensión del estado del proceso de pruebas, o mayor motivación del personal de pruebas.

Niveles.

- A. Herramientas de planificación y control.
- B. Herramientas de ejecución y análisis, y de pruebas
- C. Automatización integral de las pruebas.

AC9. Entorno de pruebas.

La ejecución de las prueba tiene lugar en el denominado entorno de pruebas. Este entorno de pruebas consta de los siguientes componentes: el hardware, el software, las instalaciones de comunicaciones, las herramientas para la creación y el uso de datos de prueba, y los procedimientos. El entorno debe quedar establecido para poder hacer pruebas de forma óptima, ya que influye en gran medida en la calidad, duración y coste

del proceso de pruebas. Algunos aspectos que también resultan importantes del entorno de pruebas son: los roles y las responsabilidades, el control, la suficiente disponibilidad, la flexibilidad y la representatividad de los entornos reales de producción.

Niveles.

- A. Entorno administrado y controlado.
- B. Pruebas en el entorno más conveniente.
- C. Montaje de entorno a solicitud.

AC10. Entorno de oficina.

El personal de pruebas necesita oficinas, escritorios, sillas, ordenadores, grabadoras, impresoras, teléfonos, etc. Un arreglo adecuado y a tiempo del entorno de oficina influye positivamente en la motivación de dicho personal, y en la comunicación y eficiencia de la ejecución de las tareas de pruebas.

Niveles.

- A. Entorno de oficina adecuado y a tiempo.

AC11. Compromiso y motivación.

El compromiso y la motivación del personal involucrado en las pruebas son condiciones indispensables para lograr un proceso de pruebas maduro. El personal involucrado en el proceso de pruebas incluye no solamente a los miembros del equipo de pruebas, sino también, entre otros, a líderes de proyectos y a los directivos. El proceso de pruebas debe disponer del tiempo, dinero y recursos suficientes (cuantitativa y cualitativamente) para efectuar una buena prueba. La cooperación y comunicación con los otros miembros del proyecto da como resultado un proceso eficiente y una involucración temprana.

Niveles.

- A. Asignación de presupuesto y tiempo.
- B. Las pruebas integradas en la organización del proyecto.
- C. Ingeniería de Pruebas.

AC12. Funciones de pruebas y de capacitación.

El personal de pruebas requiere de una cierta preparación, una combinación de diferentes materias, funciones, conocimientos y habilidades. Por ejemplo, aparte de tener experiencia específica sobre pruebas, también se requiere conocimiento del sistema que está siendo probado, conocimiento de la organización y conocimiento general de automatización. También es importante contar con ciertas habilidades sociales. Para poder tener estas cualidades, es necesario ofrecer educación y capacitación.

Niveles.

- A. Gerente de pruebas y probadores.
- B. Soporte (metodológico, técnico, funcional), control.
- C. Aseguramiento de la calidad interno.

AC13. Alcance de la metodología.

Para cada proceso de pruebas se utiliza cierta metodología o acercamiento, que consiste en actividades, procedimientos, estándares, técnicas, etc. El objetivo de una organización es utilizar una metodología que sea lo suficientemente genérica como para ser ampliamente aplicable, pero al mismo tiempo que sea lo suficientemente detallada para poder evitar las reinventaciones en cada nuevo proceso de pruebas.

Niveles.

- A. Específico al Proyecto.
- B. Para toda la organización, genérica.
- C. Organización, optimización (Investigación y desarrollo).

AC14. Comunicación.

En un proceso de pruebas, la comunicación se lleva a cabo de diversos modos, tanto entre los controladores como grupo, como entre los controladores y otros miembros del proyecto, tales como el desarrollador, el usuario final, y el líder del proyecto. Algunos temas objeto de comunicación son la estrategia de pruebas, así como el progreso y la calidad del software bajo prueba.

Niveles.

- A. Comunicación interna.
- B. Comunicación del Proyecto (defectos, cambios, control).
- C. Comunicación en la organización.

AC15. Informes.

Realizar pruebas no está relacionado solamente con la detección de defectos, sino también con informar sobre la calidad (o falta de calidad) del software. Los informes deben dirigirse a proporcionar información bien fundada hacia el proyecto y el cliente sobre la calidad del software e incluso sobre el proceso de desarrollo del software.

Niveles.

- A. Defectos.
- B. Progreso (status de pruebas y productos), actividades (costes + tiempo, metas), defectos con prioridades
- C. Riesgo y Consejo, incluyendo estadísticas.

D. SPI consejo

AC16. Gestión de defectos.

Una buena administración debería ser capaz de controlar el ciclo de vida de un defecto y crear diferentes informes (estadísticos). Estos informes se usan para prestar asesoramiento bien fundado sobre la calidad del software. La gestión de defectos es responsabilidad del líder de proyecto, aunque los probadores están altamente involucrados en ello.

Niveles.

- A. Administración interna de defectos.
- B. Administración extendida de defectos, informes flexibles.
- C. Administración de defectos del proyecto.

AC17. Administración de los elementos de prueba.

Los elementos de prueba deberían ser mantenibles y reutilizables, y por lo tanto, deben ser administrados. Aparte de los elementos de pruebas, también los productos de fases previas, tales como el diseño y la construcción, deben estar bien administrados. Una buena administración de estos productos incrementa la factibilidad de pruebas (y por ende la calidad) del software.

Niveles.

- A. Administración y control interno de los conceptos de pruebas
- B. Administración y control externo de la base y de los objetos de pruebas.
- C. Elementos de pruebas reutilizables.
- D. Capacidad de Rastreo: desde los requerimientos hasta los casos de prueba.

AC18. Administración del proceso de pruebas.

Un proceso de pruebas bien administrado es de la mayor importancia para efectuar la mejor prueba posible en la a veces turbulenta área de pruebas. Con el fin de administrar cada proceso y cada actividad, son esenciales los cuatro pasos del llamado círculo de calidad de Deming: Planear, Hacer, Comprobar, Actuar.

Niveles.

- A. Planear, Hacer.
- B. Planear, Hacer, Comprobar, Reaccionar.
- C. Comprobar, Reaccionar en la Organización.

AC19. Revisión estructurada.

Revisión Estructurada en este contexto significa validar algunos conceptos tales como el diseño funcional. En comparación con las pruebas, la ventaja de la revisión estructurada es que ofrece la oportunidad de detectar defectos con prontitud. Esto lleva a que los costes de reparación sean considerablemente más bajos. Además, realizar dicha revisión es relativamente simple dado que ningún programa debe ser ejecutado y no se requiere establecer ningún entorno de pruebas, etc.

Niveles.

- A. Técnicas de revisión estructurada.
- B. Estrategia de revisión estructurada.

AC20. Pruebas de caja blanca.

Una prueba de caja blanca se define como una prueba de las propiedades internas de un objeto, mediante el conocimiento de funciones internas. Estas pruebas son ejecutadas por los desarrolladores. La prueba unitaria y la de integración son pruebas bastante conocidas de caja blanca. Al igual que la Revisión Estructurada, estas pruebas son efectuadas en las etapas iniciales del ciclo de vida antes de llegar a las pruebas de caja negra. Por otro lado, las pruebas de caja blanca son relativamente baratas porque se requiere menor comunicación y porque el análisis es más sencillo (la persona que detecta los defectos es regularmente la misma persona que hace las reparaciones. Además, se prueban menos objetos).

Niveles.

- A. Ciclo de Vida: planear, diseñar, ejecutar.
- B. Técnicas de diseño de caja blanca.
- C. Estrategia de pruebas de caja blanca

3.2.5 TMap Next: Aproximación a la gestión de pruebas

La aproximación a la gestión de pruebas de Sogeti (TMap®) ofrece una mayor rapidez en la puesta en producción de un software de gran calidad. Tmap Next [DRIEL] es una actualización de Tmap, la cual es considerada como un estándar enfocado a las pruebas estructuradas. Desde el lanzamiento de los primeros libros basados en Tmap en 1995, esta metodología se ha ido haciendo fuerte, entre otros sectores, en la industria del software. Forma un método basado en la experiencia, de cientos de probadores en distintos proyectos, siendo el objetivo de éste reunir las mejores prácticas. Tmap se centra en la organización del proceso de pruebas y en sus probadores, en vez de centrarse en aspectos técnicos de la prueba. Se trata de una metodología del proceso de pruebas que presta especial atención a la fase de planificación y de preparación, ambas cruciales para alcanzar un proceso de prueba de alta calidad.

3.2.5.1 Estructura de la metodología

TMAP es una metodología orientada al desarrollo de pruebas software que se desarrolla como complemento de TPI, proporcionando así una herramienta muy centrada en el proceso de pruebas [KARINSALO]. Se trata de un enfoque que puede ser aplicado en todas las situaciones de las pruebas y en combinación con cualquier método de desarrollo [AALST06]. Ofrece al probador una serie de elementos para realizar pruebas como: las técnicas de diseño, la infraestructura, la estrategia, las etapas, la organización y las herramientas de pruebas, estructurados en cuatro aspectos fundamentales (ver Figura 14) [DRIEL]:

1. Gestión de pruebas impulsadas por la empresa (BDTM).
2. Proceso de pruebas estructurado.
3. Kit de herramientas.
4. Adaptabilidad.



Figura 14. Estructura de TMap Next [DRIEL].

La primera parte de ésta metodología se relaciona directamente con el hecho de que la función de negocio de las TI cada vez cobra mayor importancia en las organizaciones. Por ello, la **gestión de pruebas impulsadas por la empresa (BDTM)** establece el deber del cliente de influir en cuatro aspectos de control: los resultados, los riesgos, el tiempo y los costes [DRIEL]. Tomando el cliente parte en el seguimiento y en el control de las pruebas, de manera que pueda comprender los resultados de las mismas.

En la segunda parte de la metodología, se establece una guía para el desarrollo del **proceso de pruebas estructurado**, donde Tmap utiliza su propio ciclo de vida. Se compone de los siguientes elementos: el **plan de pruebas maestro**, cuyo principal objetivo es detectar los defectos más importantes con el menor coste y tiempo posibles; las **pruebas de aceptación y de sistema**, que facilitan que el modelo sea aplicable en todos los niveles de las pruebas, permitiéndolos coordinarse; las **pruebas de desarrollo**, las cuales forman parte integral del trabajo realizado por los programadores [KARINSALO].

Tanto para realizar el plan de pruebas maestro como para establecer los niveles de prueba, es importante organizar un proceso satisfactoriamente. Así, se podrá realizar la planificación, preparación, ejecución y gestión de las actividades. Los niveles de prueba

son establecidos por la organización, consistiendo en un grupo de actividades de prueba relacionadas.

Posteriormente, en la tercera parte, el **kit de herramientas**, Tmap Next se encuentra dotado de una gran cantidad de información práctica acerca de las infraestructuras, las técnicas y la organización que han de llevarse a cabo para realizar un proceso de pruebas correcto. Sobre las **técnicas**, la metodología se centra en cómo se realizan las pruebas. Por parte de las **infraestructuras**, se definen los diferentes entornos que afectan a las pruebas, las herramientas útiles y las condiciones donde se desarrolla el trabajo. Por último, en la **organización** se establecen los roles y las responsabilidades, las tareas y las autorizaciones que deben de tener los miembros del equipo de pruebas [DRIEL].

La organización existente configura una línea de soporte para el software independiente de las pruebas específicas o del objeto de pruebas específicas. Estableciendo para ello, un soporte general mediante las herramientas de prueba, de configuración y la gestión del entorno de pruebas. Estas actividades de soporte pueden ser organizadas como un proceso. Resumiendo, TMap presenta con éste un total de cuatro tipos de procesos: el **plan de pruebas maestro**, las **pruebas de sistema y de aceptación**, las **pruebas de desarrollo** y los **procesos de soporte** [DRIEL].

TMap además posee una manera flexible de implementarse según el sistema de desarrollo, presentando así un método **adaptable**. En él, se forma la **respuesta a los cambios** donde se obtiene información sobre el entorno en el que se ejecutan las pruebas y se establecen posibles cambios que puedan desempeñar un papel importante [KARINSALO]. Además, gracias a la **reutilización de productos y procesos** se definen las actividades para identificar la reutilización y preservación de los objetos de las pruebas. Para completar esta última parte del modelo, se desarrolla el **aprendizaje desde la experiencia**, ya que TMap® Next al ser un método ofrece el marco necesario para aprender y aplicar lo que ya se ha utilizado con anterioridad. Sin olvidar que la metodología establece la necesidad de emplear tanto un determinado tiempo como el uso de herramientas con el fin **probar antes de usar** el sistema.

Gestión de pruebas impulsadas por la empresa

Idealmente, si una organización tuviese infinitos recursos sería lógico probar todo lo que fuese posible, como esto no suele suceder se debe elegir qué y cómo probar. La elección depende en gran manera de los riesgos que quiera asumir la organización, a modo de tiempo y dinero y, el resultado que se quiera conseguir. La elección basada en riesgos, resultados, tiempo y costes es la base de la **gestión de pruebas impulsadas por la empresa (BDTM)** [AALST06].

La aproximación BDTM se base en el concepto del caso de negocio, por el cual cada vez asume mayor importancia la perspectiva económica en los proyectos IT. Tmap Next está dotado de una serie de características para satisfacer este concepto [DRIEL]:

- La aproximación se centra en conseguir un determinado resultado.
- Conseguir el resultado final en los plazos marcados.

- El coste del proyecto ha de estar equilibrado con los beneficios que la organización espera conseguir.
- Los riesgos han de ser conocidos y lo más pequeños que sea posible.

BDTM parte del principio de que el cliente debe estar implicado en el proceso de pruebas, dotando al proceso de un carácter económico [AALST06], por ello necesita cierta información basada en que [MARTIN02]: el esfuerzo total de las pruebas está relacionado con los riesgos que tienen que ser probados; y que el esfuerzo estimado y su planificación para el proceso está relacionada con la estrategia de pruebas definida.

Con el fin de desarrollar la gestión de pruebas impulsadas por la empresa, Tmap estructura éste proceso en las siguientes seis actividades (ver Figura 15) [MARTIN02]:

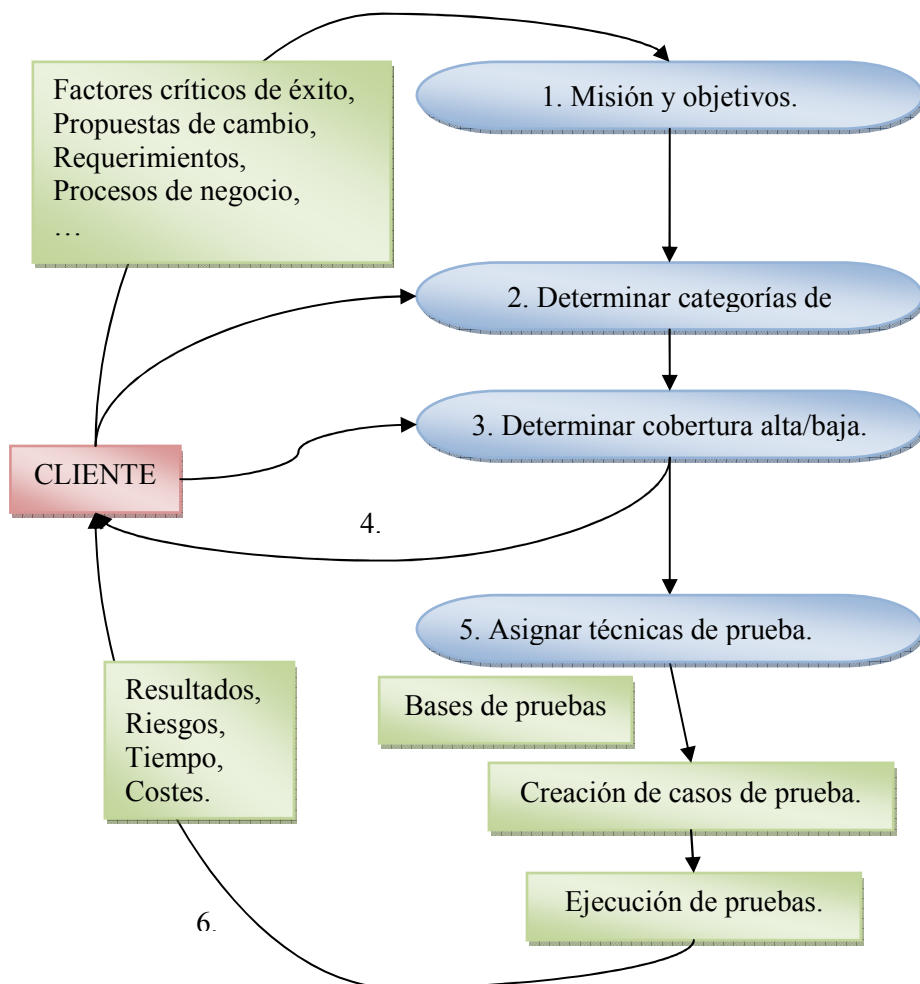


Figura 15. Proceso de BDTM

- 1. Formular la misión y recoger objetivos de las pruebas.** Realizada en primer lugar, esta actividad se lleva a cabo en contacto con el cliente. En dicha actividad, el gerente de pruebas designa la misión teniendo en cuenta los cuatro aspectos de BDTM: resultados, riesgos, tiempo y costes.

2. **Determinar las categorías de los riesgos.** Después de formular la misión, se estudia que conjunto de niveles de prueba deben crearse. Esto se realiza atendiendo al análisis de los riesgos de los productos que deben ser probados y sus características.
3. **Determinar cobertura alta/baja.** Una vez determinada la categoría del riesgo, se determina cuando un conjunto de características o una parte de un objeto debe ser probada. Además, se obtiene si se hace a fondo o por encima.
4. **Estimar globalmente las pruebas y la planificación creadas.** Una vez finalizado el paso anterior, con la aplicación de esta actividad se comunica a las partes interesadas la realización de los ajustes necesarios. Esto se produce con el fin de dotar al proceso de un balance entre los costes y los riesgos. Cuando se produzcan cambios es necesario repetir los pasos 3 y 4.
5. **Asignar técnicas de pruebas.** Cuando las distintas partes interesadas en el proceso dan el visto bueno en el paso 4, el gerente de pruebas asigna quién se ocupa de las pruebas y con qué técnicas se tratan.
6. **Proporcionar al cliente información sobre el proceso de pruebas.** Durante todo el proceso, el gerente de pruebas debe informar a las distintas partes interesadas en del progreso, los riesgos, los costes y la calidad del proceso.

Proceso de pruebas estructurado

El proceso de pruebas se realiza en todos los niveles de los distintos tipos de prueba, coordinándolos y sincronizándolos. De modo que se encarga de transmitir tareas y responsabilidades entre los distintos grupos involucrados en el proceso, para ello divide el proceso de pruebas en cuatro fases [DUTTA99]. Esto se realiza en base al propio ciclo de vida de Tmap, y estableciendo dichas fases en tres procesos de pruebas [AALST06]:

El plan de pruebas maestro: El plan de pruebas maestro (MTP) se trata de la planificación base de actividades destinadas a la detección de los defectos más importantes con el menor coste y tiempo posibles [MARTIN02]. En él, el gerente de pruebas, en consulta con el cliente y otras partes interesadas, adquiere una visión global del proceso a través de los distintos niveles de pruebas. Diferenciando que debe ser probado, cuando y con qué rigor [MARTIN02]. Constituyendo así la base para detallar los planes de prueba en los distintos niveles. El plan de pruebas maestro se compone de dos fases:

Planificación general del proceso de pruebas: El plan de pruebas maestro es realizado por el gerente de pruebas en colaboración con las distintas partes interesadas en el proceso, respetando las cuatro características básicas de BDTM. En esta fase [AALST06] se planifica el desarrollo de los distintos niveles y se establecen y se asignan los distintos objetivos, poniendo en marcha los cuatro primeros pasos de BDTM.

Control general del proceso de pruebas: En esta fase se controla el proceso, la infraestructura y los productos de las pruebas a nivel global. Esto se produce con el fin de proporcionar un avance continuo de la calidad del proceso y del objeto de prueba.

Además, se adquiere mayor importancia en la realización de informes durante todas las etapas del proceso. De esta forma, tanto los probadores como los clientes son conscientes en cualquier momento del estado de la calidad.

Las pruebas de aceptación y de sistema: Las pruebas de aceptación y de sistema son consideradas como procesos organizados de forma autónoma. Tienen un propio plan de prueba, un presupuesto y a menudo su propio entorno de prueba. Se tratan de procesos ejecutados en paralelo al proceso de desarrollo, debiendo iniciarse al mismo tiempo que se crean las especificaciones funcionales [AALST06].

Dentro del proceso de pruebas, el grupo de los clientes queda representado por la parte de aceptación, mientras que el proveedor queda representado por la parte de sistema. Cada uno por lo tanto posee sus propias responsabilidades dentro del proceso. Así, los proveedores realizan las *pruebas de sistema* con el fin de determinar el cumplimiento de las especificaciones técnicas y funcionales. Una vez realizadas estas pruebas y depurados los resultados, el sistema es puesto a disposición del cliente para las *pruebas de aceptación*. Estableciendo con estas últimas si el producto se encuentra correctamente entregado y cuándo se va a poder utilizar.

En ambas pruebas, con el fin de crear del plan de pruebas y ejecutar distintas actividades dentro del proceso se utiliza el modelo de ciclo de vida Tmap como referente [TMAP]. A continuación, se describe dicho modelo, con sus correspondientes actividades, para los distintos niveles de pruebas de aceptación y de sistema.

3.2.5.1.1 Modelo del ciclo de vida de Tmap

El modelo del ciclo de vida de TMap es un modelo genérico, pudiendo ser aplicado en todos los niveles y tipos de pruebas. Además, es posible utilizarlo en paralelo con los modelos del ciclo de vida del desarrollo de sistemas [TMAP].

El modelo del ciclo de vida TMap se divide en siete fases [DRIEL], las cuales se desarrollan a continuación:

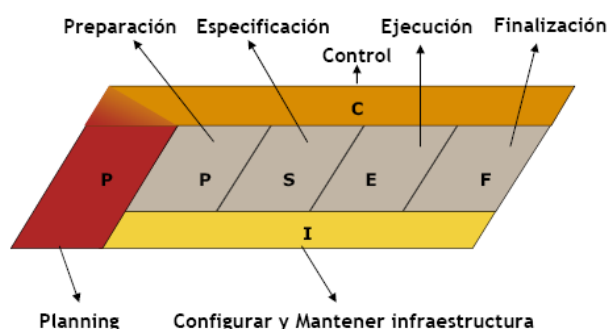


Figura 16. Modelo del ciclo de vida TMap [DRIEL].

Planificación. En la fase de planificación se formula de una forma amplia y coherente el enfoque sobre el cual se basa la asignación de las pruebas. Una parte importante de esta fase es la creación de un plan de pruebas. Éste tiene el propósito de informar al cliente y a las distintas partes interesadas sobre el planteamiento, el cronograma, el presupuesto, las actividades y los productos que deben entregarse en

relación con el proceso de prueba. Si existe un plan de pruebas maestro, el plan de pruebas se debe ser derivado de él.

Control. En la fase de control, se proporciona al cliente el conocimiento y la oportunidad de influir en el avance del proceso, en la calidad y en los riesgos del objeto de pruebas. El objetivo de las actividades en esta fase es controlar e informar de los procesos de pruebas de una manera óptima, de tal forma que el cliente tenga una visión adecuada y pueda ejercer el control del progreso y de la calidad del proceso de pruebas.

Fase de configuración y mantenimiento de la estructura. En esta fase se proporcionan las infraestructuras y los recursos requeridos por las pruebas. Se desarrolla en paralelo con las fases de preparación, especificación, ejecución y finalización de Tmap.

Preparación. En la fase de preparación se realiza la revisión de las propias pruebas. El principal objetivo de esta fase es tener acceso a las bases de las pruebas, con el fin de adecuar la calidad a los diseños de las pruebas que han sido acordados con los clientes.

Especificación. La fase de especificación detalla las pruebas requeridas y la situación de partida, preparando las pruebas que puedan ser ejecutadas tan rápido como sea posible una vez los desarrolladores obtienen un objeto de prueba.

Ejecución. En ésta fase se obtiene la visión de la calidad del objeto de prueba a través de la ejecución de las pruebas de conformidad.

Finalización. En la fase de finalización y gracias a la estructura de la aproximación de TMap se da la posibilidad a los productos de ser reutilizados en pruebas posteriores, permitiendo acelerar ciertas actividades. Los productos pueden ser tanto resultados tangibles como casos de prueba, pero también intangibles como la experiencia (evaluación del proceso).

FASE	ACTIVIDADES
FINALIZACIÓN	Evaluar el proceso de pruebas. Preservar los casos de prueba.
EJECUCIÓN	Entrada del objeto de pruebas. Preparar las situaciones iniciales. Ejecutar las pruebas. Comprobar y evaluar los resultados.
ESPECIFICACIÓN	Crear las especificaciones de las pruebas. Definir las situaciones iniciales. Especificar la entrada del objeto de prueba.
PREPARACIÓN	Coleccionar la base de pruebas. Crear listas de control. Evaluar la base de pruebas. Crear el informe de las pruebas.

FASE	ACTIVIDADES
CONFIGURACIÓN Y MANTENIMIENTO.	Especificar la infraestructura. Realizar la infraestructura. Especificar la validación de la infraestructura. Validar la infraestructura. Mantener la infraestructura. Preservar la infraestructura.
CONTROL	Gestión. Seguimiento. Redactar informes. Realizar ajustes.
PLANIFICACION	Establecer la misión. Entender la misión. Determinar la base de las pruebas. Analizar los riesgos del producto. Determinar la estrategia de las pruebas. Estimar esfuerzos. Planificar el proyecto. Asignar unidades de pruebas y técnicas de pruebas. Definir los productos de las pruebas. Definir la organización. Definir la infraestructura. Organizar la gestión. Determinar riesgos de proceso y definir las medidas. Retroalimentación y consolidación del plan.

Tabla 13. Tabla de relación Actividad/Fase del ciclo de vida de Tmap.

Las pruebas de desarrollo: Las pruebas de desarrollo se conocen como las pruebas en las se utiliza el conocimiento de la ejecución técnica del sistema [TMAP]. Se comienzan con las pruebas de las partes más pequeñas del sistema, es decir: las rutinas, las unidades, los programas, los módulos y los objetos. Después de que se hayan establecido las partes más elementales del sistema con una calidad aceptable, el sistema es sometido a pruebas integrales.

Las *pruebas unitarias* son términos utilizados exclusivamente en este contexto. Éstas son llevadas a cabo en el entorno de desarrollo, con el objetivo de demostrar que una unidad satisface los requerimientos definidos por la especificación técnica. La integración de las unidades de prueba es llevada a cabo por desarrolladores.

A pesar de ser un proceso de pruebas realizado a nivel de desarrollo, se identifican y describen una serie de actividades basadas en el ciclo de vida modelo TMap [MARTIN02] orientándose al nivel de desarrollo, a excepción de las tres actividades descritas a continuación.

Desarrollar las unidades de prueba (TDD): Esta actividad se produce en el caso de utilizar el desarrollo dirigido de pruebas (TDDI). El cual es un método de desarrollo que

tiene gran influencia sobre las pruebas de unidad, cuyo objetivo es conseguir reacciones más rápidas. Es una de las mejores prácticas de la programación extrema (XP).

Ejecutar una revisión de código (opcional): Durante, o inmediatamente después, de la ejecución de la unidad de prueba se debe realizar una revisión de código. Estableciendo que operaciones se deben llevar a cabo de acuerdo con las guías de desarrollo.

Preparar el siguiente nivel de pruebas: Esta actividad consiste en acondicionar el actual sistema de pruebas al siguiente nivel, esto se puede realizar de varias maneras. La forma ideal de hacerlo consiste en una demostración, en la cual el desarrollado demuestra que el sistema funciona en el entorno de pruebas.

Conjunto de herramientas

TMap ofrece un buen soporte para la ejecución del proceso pruebas estructuradas a través de un completo conjunto de herramientas [AALST06]. Este conjunto de herramientas se basa en el trabajo a través de técnicas (cómo se prueba), infraestructuras (dónde y con qué) y organización (quién).

Técnicas: Una técnica de pruebas es una combinación de acciones para producir un producto de prueba de manera universal [AALST06]. TMap ofrece diferentes técnicas con el fin de solventar las siguientes necesidades.

- Estimación de pruebas.
- Gestión de defectos.
- Crear métricas.
- Análisis de los riesgos del producto.
- Diseño de pruebas.
- Evaluación del producto.
- Diferentes listas de control.

TMap ofrece una gran variedad de listas de control que son de utilidad al probador en determinadas actividades. Por ejemplo, hay listas de comprobación que puede ser utilizado como apoyo en el balance de la misión, determinación de las instalaciones de pruebas, determinar los riesgos de prueba del proyecto, el establecimiento de la estrategia de prueba, la evaluación del proceso de prueba, realizando entrevistas y determinar si la información adecuada está disponible para utilizar una técnica específica de diseño del ensayo [TMAP].

Infraestructura: Con el fin de ejecutar el proceso de prueba es necesario establecer cierta infraestructura que se encargue de ello [DUTTA99]. Parte de la infraestructura viene dada por el **entorno de prueba**. El cuál se trata de un sistema de componentes tanto hardware como software (interfaces, datos del entorno, herramientas de gestión y

procesos) en el que se ejecuta una prueba. La configuración y composición de este entorno depende del objetivo de la prueba.

Una buena infraestructura se encuentra basada en el uso de **herramientas de pruebas**. Éstas se tratan de instrumentos automáticos que proporcionan soporte a una o más actividades de pruebas, tales como la planificación y el control, la especificación de la prueba, y la ejecución de la misma [TMAP].

Por último, para definir completamente la infraestructura de las pruebas se deben estudiar los **lugares de trabajo**. Con esto, se trata de establecer la disponibilidad de un lugar de trabajo donde los probadores puedan realizar su trabajo en buenas condiciones, de manera eficaz y eficiente [TMAP].

Organización: Los procesos de prueba que no están debidamente organizados suelen tener resultados desastrosos. Por ello, la participación de disciplinas muy distintas en los proyectos genera conflictos de intereses, imprevisibilidad y una gestión compleja de tareas [AALST06]. De esta forma, una organización de las pruebas puede establecer la creación de relaciones efectivas entre las funciones, las instalaciones y las actividades de las pruebas con el fin de mejorar de manera paulatina la calidad.

Para establecer una buena organización de las pruebas estructuradas es necesario establecer los **procesos de soporte**. En estos procesos se hace especial hincapié en **políticas de prueba** y en la **organización de las pruebas en los proyectos**. Donde se definen los roles, las tareas, las responsabilidades y las autorizaciones que van a estar presentes en un determinado proyecto.

Adaptabilidad

La capacidad de adaptación de TMap se centra en que puede ser utilizado en cualquier situación y en entornos cambiantes [DUTTA99]. Esta capacidad se inicia desde las primeras actividades del plan de pruebas maestro, con la determinación de los cambios y respondiendo a los mismos. Esta adaptación a los cambios se conoce como la **respuesta a los cambios**, por la cual si la estrategia, la estimación y la planificación no son aceptadas por el cliente, el plan se adapta. De esta forma, se otorga al cliente el control del proceso de prueba y le permite realizar una gestión basada en el equilibrio entre el resultado y los riesgos, por una parte, y del tiempo y el coste por otra [TMAP].

Además de la respuesta a los cambios, es necesario con el fin de fomentar la capacidad de adaptación **reutilizar los productos y los procesos**. [DRIEL]. El énfasis en este aspecto se encuentra en la fase de finalización, donde las actividades se definen para identificar lo que puede ser reutilizado y cómo pueden ser óptimamente conservadas.

Como método, TMap ofrece la posibilidad de **aprender de la experiencia**, para lo cual se encuentra dotado del espacio para aprender y aplicar lo utilizado con anterioridad. Por ello, se evalúa el propio proceso de prueba, adquiriendo gran importancia el uso de métricas orientadas a dicho fin. Su uso facilita la gestión del proceso de prueba, justificando las recomendaciones de las pruebas y comparando los sistemas o los procesos [TMAP].

Por último, para completar la adaptabilidad de Tmap el modelo recoge la posibilidad de **probar antes de usar**, creando un espacio para realizar pruebas previas a la puesta en funcionamiento del producto. Las principales herramientas para dichas pruebas son actividades relacionadas con la entrada de datos.

Procesos de soporte: A menudo, resulta más eficiente centralizar los procesos de soporte de las pruebas que implementarlos a nivel de proyecto [TMAP]. En Tmap, con el fin de organizar el soporte de pruebas se atiende a las siguientes necesidades: la política de pruebas, la organización permanente de las pruebas, el entorno de las pruebas, las herramientas de las pruebas.

Políticas de las pruebas: Las políticas de las pruebas describen como una organización se relaciona con sus empleados, recursos y metodologías, teniendo en cuenta las distintas situaciones que se pueden dar dentro de un proceso de pruebas [DRIEL]. Por esta razón, Tmap recomienda su correlación con otros tipos de políticas como [TMAP]:

- **Política estratégica.** La política estratégica de una organización tiene un gran impacto en sus componentes y actividades fundamentales, incluidas las pruebas. En ella se definen condiciones necesarias para realizar la organización interna y cumplir con los objetivos de calidad.
- **Política operacional.** Esta política atiende tanto al soporte como a la ejecución del proceso de pruebas. Cuando la política operacional atiende a temas de soporte aporta características técnicas, metodológicas y funcionales a los probadores. Sin embargo atendiendo a la ejecución define pruebas y la gestión de las mismas de forma independiente para cada organización.
- **Política táctica.** La política táctica se traduce en la implementación de la política operacional. En ella se describe cómo debe de ser configurado el contenido del enfoque estructurado de las pruebas en una organización.

Organización de pruebas permanente: El uso de una organización de pruebas permanente establece una estructura común a distintos proyectos [DRIEL]. En esta forma de organizar el proceso de pruebas, influye la necesidad de mejorar los conocimientos en determinados casos como: la normalización de los productos, la limitación del inicio del proyecto, la mejora continua del proceso, la consolidación de experiencias, y el conocimiento de los costes y el tiempo de entrega de las pruebas.

Entorno de pruebas: A la hora de establecer el proceso, se debe crear y configurar un entorno de pruebas en función de los objetivos marcados y ajustándose a sus pruebas dinámicas. Además, se deben formular una serie de requisitos genéricos, los cuales se deben cumplir por el entorno de pruebas con el fin de garantizar la ejecución fiable de las pruebas. De tal forma, que resulte representativo, manejable, flexible y garantice la continuidad de la ejecución de las mismas. Para prevenir problemas en el entorno de pruebas, es necesario establecer procesos de gestión de la configuración y de mantenimiento de los entornos, definiendo [DRIEL]:

- **Gestión de la configuración.** El objetivo de la gestión de la configuración es diferenciar la configuración de distintos entornos al mismo tiempo. Pudiendo

asociar así, la capacidad de cambio o el establecimiento de defectos claramente en cada uno.

- **Gestión del cambio.** El objetivo de la configuración del cambio consiste en implementar todos los cambios de forma individual, los cuales a las distintas partes interesadas les conveniente realizar. Los cambios se deben producir a consecuencia de defectos, modificaciones hardware, la implementación de nuevo software o el cambio de datos.
- **Gestión de lanzamientos.** La gestión de la puesta en marcha de un producto en la implementación de uno o más combinados y aprobados, vía gestión del cambio, cambios de un entorno a otro.

3.2.5.2 Componentes de TMap Next

En esta sección Tmap Next describe cierto número de componentes que pueden llegar a ser utilizados de forma independiente, como las características de calidad y los roles de prueba. También, pueden ser utilizados como complemento de los procesos, por ejemplo con la utilización del análisis de los riesgos del producto y las técnicas diseño de pruebas [AALST06].

Análisis de los riesgos de un producto.

Las pruebas son unas medidas que ofrecen una visión de los productos desarrollados y de los riesgos relacionados en su producción. Pero, a la para dichas mediciones no existen recursos y tiempo ilimitados. Por ello, es importante relacionar el esfuerzo de las pruebas con los riesgos previstos, tomando como lema “si no hay riesgo, no se prueba” [AALST06]. En este contexto cobran especial importancia las elecciones bien justificadas, utilizando el **análisis de los riesgos del producto (PRA)** [AALST06] como una herramienta que facilita la toma de decisiones.

El PRA requiere un marco común de referencia para llevarse a cabo y de esta forma comprender lo que representa. Tmap establece el **riesgo del producto** como la posibilidad de que un producto falle en relación con el daño esperado si esto ocurre. La posibilidad de fallo a su vez viene dado por la posibilidad de que ocurra un defecto por su frecuencia de uso.

Características de calidad.

Tmap utiliza una serie de características de calidad con el fin de realizar las pruebas. Aunque a nivel internacional existen otros conjuntos como el estándar ISO9126. La utilidad de cualquier conjunto de características es recomendada para completar las pruebas.

Técnicas.

Tmap propone distintas técnicas para poder llevar a cabo un desarrollo correcto del proceso de pruebas, entre las que destacan las estimaciones, los diseños y la evaluación.

Las estimaciones se realizan en diferentes niveles del proceso de pruebas. A continuación se muestran dichos niveles donde pueden aparecer reflejadas [TMAP]:

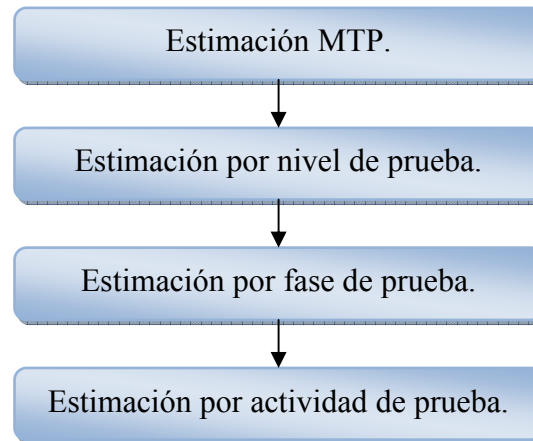


Figura 17. Niveles de las técnicas de estimación.

Independientemente del nivel donde se realicen dichas técnicas, con el fin de crear estimaciones consistentes se han de seguir los siguientes pasos [DRIEL]:

- Controlar el material que puede servir de base para la estimación.
- Seleccionar técnicas de estimación.
- Determinar la estimación final.
- Presentar los resultados.

Existen diversas técnicas de estimación propuestas por Tmap que deben elegirse correctamente para poder realizar la estimación en cada nivel.

Otra técnica destacada por Tmap se trata de la **técnica de diseño de prueba** que se utiliza para derivar los casos necesarios que permitan alcanzar la cobertura necesaria de una base de pruebas concreta. La técnica se encuentra estrechamente relacionada con el tipo de prueba que se ejecuta y con ello, la calidad característica de que se prueba con el tipo de prueba [AALST06].

Una técnica de diseño de pruebas es un método estandarizado destinado a realizar cálculos. Éstos se realizan a partir de una base específica de casos de prueba, los cuales ofrecen una cobertura específica. Por ello, una técnica de diseño de pruebas describe la base de la prueba requerida, los tipos de cobertura prevista, y las técnicas básicas aplicadas para lograr un determinado tipo de cobertura [TMAP]. En Tmap Next se describen 11 tipos de cobertura.

Por último se establece la **técnica de evaluación**, que valora los documentos creados por otra persona. Por lo general, estos resultados se generan en una lista destinada al autor del documento en cuestión. Como es evidente, estos resultados dependen en gran medida de la actitud del evaluador. Así, dependiendo de la manera en que se registran o comunican las evaluaciones, el autor podría sentirse atacado. Es importante ser consciente de que el objetivo final es preparar el mejor producto posible con la colaboración de distintos agentes [AALST06].

Las técnicas de evaluación resultan muy adecuadas para mejorar la calidad de los productos. Aunque, no solo sólo son aplicables a los productos de la propia organización. TMap describe y utiliza para ello inspecciones, revisiones y walkthroughs.

Gestión de defectos.

Aunque la administración y la supervisión de los defectos no es cuestión de los probadores, éstos están generalmente muy estrechamente ligados. Una buena administración debe ser capaz de controlar el ciclo de vida de un defecto y ofrecer distintas visiones generales. Esas visiones se utilizan, entre otras cosas, para hacer declaraciones de calidad bien fundadas [TMAP].

Los probadores deben tener en cuenta que se encuentran juzgando el trabajo de otras personas y que el producto final es el resultado de la colaboración entre todas las partes implicadas en el mismo. Por ello, el probador debe adoptar una postura lo más neutral posible para encontrar defectos.

En el proceso de pruebas, los probadores informan de la presencia de defectos y comprueban sus soluciones. Pero, es el gerente de pruebas quien se encarga de interactuar con el grupo implicado en el defecto, preservando así la independencia entre los dos grupos [TMAP].

Desde Tmap se propone un ciclo de vida del defecto que permite obtener estructuras y planificar su tratamiento, se compone de las fases que se presentan a continuación por orden de ejecución [AALST06]:

- Encontrar un defecto.
- Informar del defecto.
- Tratamiento de defectos.

Métricas.

La definición, el mantenimiento y el uso de métricas son importantes para el proceso de pruebas, ya que permite al gerente de pruebas comunicar el estado de éstas a través de hechos [MARTIN02]. TMap ofrece instrucciones de cómo establecer un conjunto práctico de iniciación a las métricas de las pruebas basado en el método de las seis etapas de métrica-pregunta-objetivo, las cuales son:

- Definición de objetivos.
- Realizar preguntas sobre los defectos por objetivo.
- Realizar preguntas sobre las posibles métricas a aplicar.
- Recoger información del proceso de pruebas y analizar.
- Presentar y distribuir los datos de medida.
- Relacionar los datos de medida con las preguntas y objetivos.

Además, TMap proporciona listas de métricas útiles para hacer valoraciones sobre la calidad del objeto y del proceso de prueba [AALST06]:

- **Números de defectos encontrados.** Ratio entre el número de defectos encontrados y el tamaño del sistema por unidad del tiempo de prueba.
- **Instrucciones ejecutadas.** Proporción del número de instrucciones del programa probadas por el número total de instrucciones.
- **Pruebas de número.** Proporción entre las pruebas y el tamaño del sistema. Indica cuantas pruebas son necesarias en una parte.
- **Numero de caminos probados.** Proporción entre el número de rutas probadas y su total.
- **Numero de defectos durante la producción.** Esta métrica indica el número de defectos no encontrados durante el proceso de prueba.
- **Efectividad de la detección de defectos.** Número total de defectos encontrados durante las pruebas, dividido entre el total de defectos estimados.
- **Coste de las pruebas.** Proporción entre los costes de las pruebas y los costes del desarrollo.
- **Coste por defecto localizado.** Coste total de las pruebas entre el número total de defectos encontrados.
- **Presupuesto utilizado.** Proporción del presupuesto y el coste actual de las pruebas.
- **Eficiencia de las pruebas.** Número de pruebas requeridas entre el número de defectos localizados.
- **Grado de automatización de las pruebas.** Proporción entre el número de pruebas llevadas a cabo de forma manual y el número de pruebas automáticas.

- **Numero de defectos encontrados (relativos).** La proporción entre el número de defectos encontrados y el tamaño del sistema por unidad del tiempo de prueba.
- **Defectos como resultados de las modificaciones no probadas.** Defectos a causa de las modificaciones que no están probadas, como una parte del total del número de defectos derivados como resultados de los cambios.
- **Defectos después de las modificaciones probadas.** Defectos a causa de las modificaciones que están probadas, como una parte del total del número de defectos derivados como resultados de los cambios.
- **Ahorros de las pruebas.** Indicador de los ahorros provocados por las pruebas.

Roles de prueba

Las actividades de prueba dentro de un proyecto o en una organización se realizan a través de distintas tareas. Para la ejecución de dichas tareas, se requiere un conocimiento y unas habilidades en particular. El conjunto de tareas junto al conocimiento y las habilidades necesarias para llevarlas a cabo se conoce como un rol [MARTIN02]. Existen roles que se corresponden con las posiciones mencionadas anteriormente como [AALST06]:

- Tester.
- Programador de herramientas de pruebas.
- Experto en métodos de prueba.
- Coordinador de pruebas.
- Experto en herramientas de pruebas.
- Consultor de pruebas.
- Gerentes de pruebas.
- Consultor de herramientas de pruebas.

Por el contrario existen otro tipo de roles que no se asemejan a ninguna posición por ellos mismos. Además de los roles descritos aquí, en la práctica se pueden necesitar otros, pudiendo ser combinados para resolver estas necesidades. En Tmap se distinguen los siguientes roles sin asemejarse a ninguna posición [AALST06]:

- Integrador de aplicaciones.
- Administrador de defectos.

- Experto en el conocimiento del dominio.
- Intermediario.
- Experto de sistemas.
- Coordinador de infraestructuras de pruebas.
- Administrador del proyecto de pruebas.
- Líder del equipo de pruebas.
- Administrador de los programas de pruebas.

3.2.6 La organización de las pruebas convencional

El desarrollo de una *organización convencional* centrada en la calidad de un proyecto no necesita una asignación específica de recursos. De hecho, hoy en día existen un gran número de empresas donde el mismo individuo programa y prueba el código [CRAIG02]. A pesar de no gozar de una buena fama, al no usar probadores profesionales, se sigue utilizando mayoritariamente en organizaciones formadas por pequeños grupos de trabajo.

Gracias a esta simple estructura, se eliminan los problemas de comunicación con los grupos de pruebas y se facilita la toma de decisiones. Además, se dota al proceso de pruebas de un mayor conocimiento del código y del diseño [TINN01].

Sin embargo, el principal perjuicio con el que cuenta esta organización se trata de la pérdida de imparcialidad entre la toma de decisiones relativas al desarrollo y a la calidad. Aparte de posible desequilibrio, existen otros factores contrarios a este tipo de organización, como son: la pérdida de especialización de los profesionales, debido a que al desarrollar tantas labores se puede perder calidad en el trabajo realizado en alguna de ellas; o el poco entendimiento del concepto de negocio del software por parte de éstos [BLACK03]. Por todo ello, el método convencional es considerado como un tipo de organización que debe implantarse bajo cierto tipo de consideraciones especiales [CRAIG02]:

- La necesidad de definir y de seguir un riguroso proceso de pruebas.
- La asignación del suficiente tiempo al proceso de pruebas.
- La existencia de experiencia en el negocio.
- La necesidad de fomentar la gestión de la configuración del software.
- La formación de los desarrolladores de las pruebas.
- El establecimiento y el seguimiento de un criterio final o de salida.

- La necesidad de un seguimiento continuo.

Ventajas
La necesidad de pocos recursos. La existencia de reportes rápidos y directos. La familiarización y el compromiso con el proyecto.
Inconvenientes
La falta de objetividad. La falta de especialización. El conflicto de intereses entre el desarrollo y las pruebas.

Tabla 14. Ventajas e inconvenientes del método convencional.

3.2.7 Los grupos SQA

El **control y la garantía de la calidad (QA)** son actividades esenciales en cualquier negocio que elabore productos de consumo. Antes del siglo XX, este proceso era responsabilidad exclusiva del empresario que fabricaba un producto. La primera función normalizada de QA la introdujeron los Laboratorios Bell en 1916, y se expandió rápidamente a través del mundo industrial. Durante la década de 1940 surgieron enfoques cada vez más formales orientados al control de la calidad, los cuales se apoyaban en la medición y la mejora continua de los procesos, como elementos clave de la gestión de la calidad [PRESS05].

En la actualidad, toda compañía posee mecanismos que garantizan la calidad de sus productos. De hecho, destacar la preocupación de una compañía por la calidad se ha convertido en una táctica de mercado durante las décadas pasadas.

La historia de la **garantía de la calidad en el desarrollo software (SQA)** ha avanzado de forma paralela a la de la calidad en la fabricación hardware [CSAE]. Durante los primeros años de existencia de las tecnologías de la información (1950,1960), la calidad era responsabilidad exclusiva del programador. Los estándares de garantía de calidad para el software se introdujeron en los contratos militares de desarrollo durante la década de 1970, y se han extendido rápidamente en la elaboración del software de los negocios.

Tal ha sido su desarrollo, que con su evolución han surgido numerosos y muy diversos participantes con responsabilidad, como: los ingenieros de software, los gerentes del proyecto, los clientes y los vendedores. Estableciéndose el grupo de SQA, estudiado a continuación, como el representante en casa del cliente [CSAE]. Es decir, las personas que realizan el SQA deben observar el software desde el punto de vista de los receptores del mismo. Formando para ello, equipos especializados en labores de aseguramiento de la calidad software, los cuales cuentan con los recursos necesarios para llevar a cabo todas sus tareas, incluyendo: la coordinación del control, la gestión del cambio y la ayuda para recopilar y analizar métricas del software [PRESS05].

3.2.7.1 El aseguramiento de la calidad del software

El objetivo del *aseguramiento de la calidad software (SQA)* es el de proporcionar a la dirección una perspectiva adecuada del proceso que utiliza el proyecto software, y de los productos a desarrollar [AMESC02]. Esta labor se realiza en concordancia con los requisitos y los estándares de desarrollo explícitamente establecidos, y las características implícitas que se esperan de cualquier software desarrollado de forma profesional [PRESS05].

En la actualidad, existe una gran variedad de definiciones acerca del SQA, definiciones de fuentes tan importantes como el IEEE, el ISO o el SEI entre otras. Aun así, todas ellas se basan en la premisa de que el SQA es un proceso que asegura la adherencia a los estándares, así como el cumplimiento de los requisitos del software. En el análisis realizado en la publicación *Software Quality Assurance* [OWEN], se recogen todas estas definiciones con el fin de unificarlas. Estableciendo finalmente el SQA como un proceso bien definido y repetible, el cual se encuentra definido en la gestión del proyecto y en el ciclo de vida del software, con el fin de realizar la revisión de los mecanismos de control interno y de asegurar la adherencia a los estándares y procedimientos software. De esta forma, se cumple con los objetivos de asegurar el cumplimiento de los requisitos, de reducir los riesgos, de evaluar los controles internos y de mejorar la calidad respetando la planificación marcada y las restricciones presupuestarias.

El proceso se encuentra compuesto por distintas fases, las cuales a su vez se encuentran divididas en una serie de actividades, cuyo desarrollo es tarea de un grupo especializado. Tal y como se ha mencionado con anterioridad, existen numerosos y diversos participantes con responsabilidad en el proceso. Adquiriendo especial importancia en su elaboración los ingenieros del software, ya que asumen responsabilidades sobre la realización del trabajo técnico; y el grupo SQA, responsable de la planificación, análisis e informes del SQA [PRESS05].

El aseguramiento de la calidad del software posee su propio ciclo de vida, en él se abarcan todas las facetas de un proyecto de software. Dicho ciclo es realizado de forma paralela a las fases de la *gestión del ciclo de vida del proyecto (PMLC)* y al *ciclo de vida del desarrollo (SDLC)*, interactuando con cada fase de estos. De esta forma, se proporciona información respecto a las cuestiones o problemas surgidos durante la realización de las actividades del SQA y se garantiza la mejora continua de las mismas [OWEN].

El SQA resulta más efectivo cuando su flujo de comunicación se realiza través de un equipo de gestión independiente, de tal forma que pueda mantener su compromiso con el proceso y sus objetivos para la entrega [PRESS05]. La correcta y completa implementación del mismo, ayuda a eliminar defectos y, a prevenir ambigüedades y cambios en los requerimientos, mejorando la satisfacción del cliente. Además, la aplicación de un proceso integrado de SQA tiene el potencial de generar el ahorro de costos a largo plazo. Esto se debe a que el coste de encontrar y reparar los defectos o problemas se incrementa significativamente con el progreso del proyecto, desde la fase de requerimientos a la fase de implementación.

Actividades del SQA

El SEI recomienda un conjunto de actividades SQA que abordan la planificación, la supervisión, la conservación de registros y el análisis y la elaboración de informes del aseguramiento de la calidad [GILL05]. También, se recomienda que dichas actividades sean realizadas por grupos especializados, normalmente grupos de SQA. A continuación, se describen las actividades de SQA recomendadas por el SEI [AMESC02] [PRESS05]:

Preparar un plan SQA para un proyecto: El *plan de SQA* se desarrolla durante la planificación del proyecto, y es revisado por todos los implicados en el mismo. En él, se rigen las actividades asociadas con la garantía de la calidad, las cuales deberán elaborar las personas a su cargo. Identificando para ello: las evaluaciones, las auditorías y las revisiones que se han de llevar a cabo, además de los estándares aplicables al proyecto y los procedimientos para el informe y el seguimiento de errores. Por último, también se establecen los documentos que debe producir el personal encargado de dichas tareas y cómo debe interactuar con el equipo de desarrollo software. El plan funciona como una plantilla para las actividades de SQA que se instituyan para cada proyecto de software.

Participar en el desarrollo de la descripción del proceso de software: El equipo de desarrollo debe seleccionar un proceso para elaborar el trabajo, pero la descripción de dicho proceso debe ser revisada por el personal de SQA. Esto se realiza con el fin de que el proceso concuerde con las políticas organizacionales, los estándares internos, los estándares impuestos de manera externa y por otras partes del plan de proyecto del software.

Revisar las actividades de ingeniería del software: El personal encargado de las labores de SQA debe identificar, documentar y seguir las desviaciones del proceso, y verificar que se hayan realizado las correcciones necesarias con el fin de comprobar su ajuste al proceso de software definido.

Auditar los productos software seleccionados: El personal encargado de las labores de SQA debe revisar los productos seleccionados. Identificando, documentando y siguiendo las desviaciones, además de verificar que se hayan realizado las correcciones necesarias. Para completar dicha revisión, se ha de informar periódicamente de los resultados del trabajo al gestor del proyecto. Esto se realiza con el fin de comprobar que los productos obtenidos se ajustan a lo planificado como partes del proceso del software.

Garantizar la documentación y la gestión de las desviaciones: Las desviaciones son variaciones conforme a lo planificado. Se pueden encontrar en el plan del proyecto, en la descripción del proceso, en los estándares aplicables o en los productos técnicos. Cuando se producen éstas variaciones, el personal encargado de las labores de SQA debe tratarlas y recoger su existencia.

Registrar cualquier falta de ajuste: El personal encargado de llevar a cabo el SQA debe de informar de la presencia de elementos que no se ajustan a lo planificado, elaborando un seguimiento de los mismos hasta su resolución.

El ciclo de vida del SQA

El proceso de SQA posee su propio ciclo de vida, el cuál cubre todas las facetas del proyecto software. Con el fin de asociarlo al proyecto, se combina con el *ciclo de vida de la gestión del proyecto (PMLC)* y el *ciclo de vida del desarrollo de sistemas (SDLC)*. Dichos ciclos están formados por ocho fases, que como se ha comentado con anterioridad, se corresponden con fases del ciclo de SQA [OWEN].

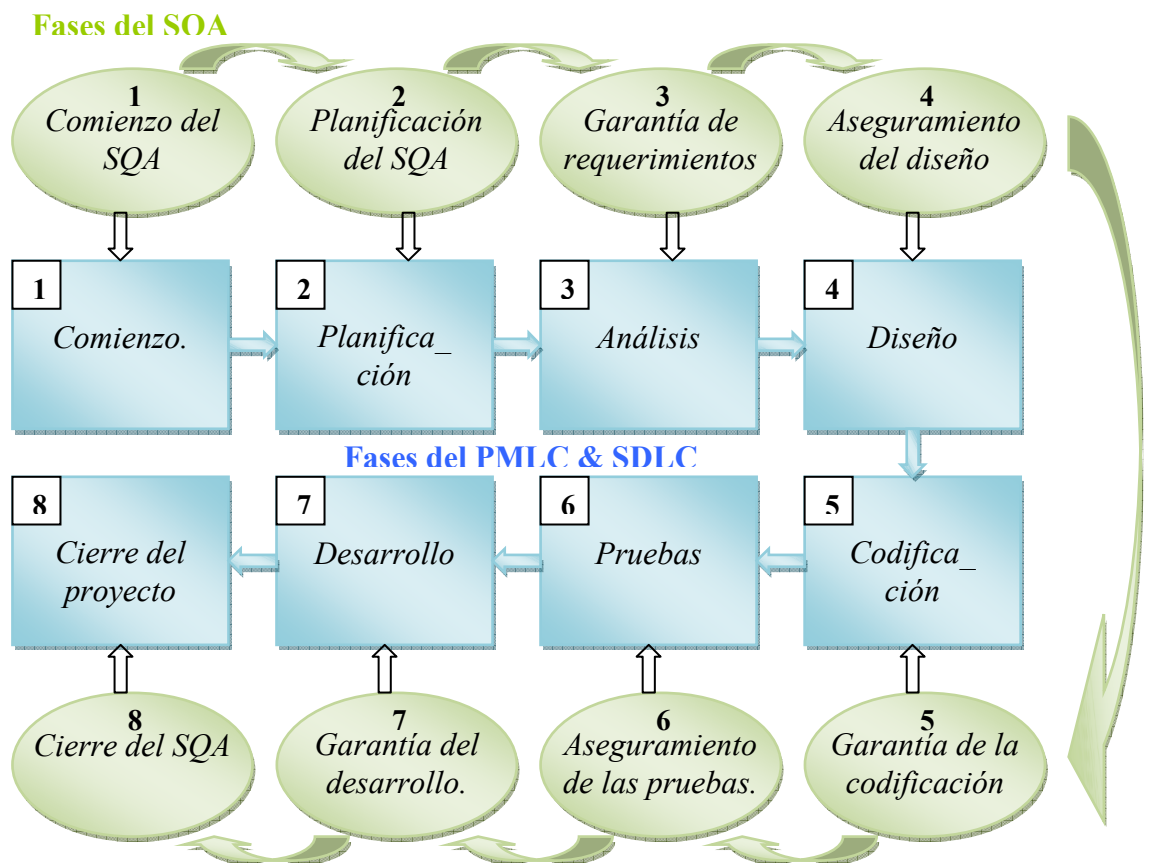


Figura 18. Ciclo de vida del SQA.

3.2.7.1.1 Fases del ciclo de vida PMLC & SDLC

1. Comienzo del proyecto: La fase de comienzo del proyecto implica la puesta en marcha de los procesos que pueden dar lugar a la aprobación del mismo, a través de una definición formal de objetivos [PAN06].

2. Planificación del proyecto: La fase de planificación del proyecto implica la definición de sus metas y objetivos, así como la planificación del curso de acciones necesarias para su cumplimiento. Como resultado, se obtienen los procesos y los procedimientos que deberán seguirse durante el proyecto [PAN06] [ALAM].

3. Análisis: La fase de análisis establece con exactitud el planteamiento de un problema mediante la recopilación y el análisis de los requisitos, creando así el modelo inicial de sistema del software [DAVIS99] [GALIN04].

4. Diseño: La fase de diseño es un proceso iterativo en el que los requisitos se traducen en una arquitectura o en un modelo para el desarrollo del software. Además, se definen las entradas, las salidas y los procedimientos de transformación del sistema [PRESS05] [GALIN04].

5. Desarrollo: La fase de desarrollo consiste en generar, revisar, compilar y probar el código [PRESS05].

6. Prueba: La fase de pruebas trata de descubrir la mayor cantidad de errores del software como sea posible, a fin de lograr un nivel aceptable de aseguramiento de la calidad [GALIN04].

7. Implementación: La fase de implementación se realiza después de que el software haya sido aprobado, y se instala en el entorno objetivo de producción [GALIN04].

8. Fin del proyecto: La fase de fin de proyecto consiste en los procesos realizados con el fin de finalizar todas las actividades de un proyecto completo o, de un proyecto cancelado [PAN06].

3.2.7.1.2 Fases del ciclo de vida de SQA

La implementación de una fase del ciclo de vida de SQA ayuda a eliminar defectos, y a prevenir ambigüedades y cambios en los requisitos, mejorando así la satisfacción del cliente. Cada fase del SQA se retroalimenta de su correspondiente fase de PMLC o de SDLC, proporcionando información respecto a las cuestiones o problemas encontrados durante las actividades de SQA, y garantizando una mejora continua. Además, en cada una de ellas se establecen salidas específicas que proveen de controles de calidad y de documentación al ciclo de vida. A continuación, se resumen las distintas áreas del ciclo de vida del SQA [OWEN]:

- 1. Comienzo del SQA:** El comienzo de las actividades de SQA se produce una vez dado el visto bueno al proyecto.

Entradas
Análisis de viabilidad. Documento de caso de negocio.
Salidas
Carta de proyecto. Declaración de los objetivos preliminares.
Controles
La verificación de que se han llevado a cabo las salidas necesarias. La comprobación de que el contenido de las salidas es correcto, consistente y completo.

Tabla 15. Elementos y actividades de la Fase 1 del SQA.

2. Planificación del SQA: La planificación del SQA define de las metas y de los objetivos del plan de SQA. Como resultado, se proporciona la especificación de los procesos o de los procedimientos de calidad que deben seguirse.

Entradas
Carta de proyecto. Normas y procedimientos del producto.
Salidas
Plan de gestión del proyecto. Métricas y plan de SQA. Plan de proyecto que representa el núcleo de las áreas de conocimiento: Plan de gestión de objetivos del proyecto, Plan de gestión de costes, Plan de comunicación, Plan de gestión de riesgos, Plan de procedimientos. Plan de gestión del cambio. Jerarquía de trabajo (WBS) y diccionario WBS. Calendario de proyecto con recursos necesarios. Roles y responsabilidades.
Controles
La verificación de que se han llevado a cabo las salidas necesarias. La comprobación de que el contenido de las salidas es correcto, consistente y completo.

Tabla 16. Elementos y actividades de la Fase 2 del SQA.

2. Aseguramiento de requisitos: La fase de aseguramiento de los requisitos garantiza que los requisitos se pueden probar y que son completos.

Entradas
Normas y procedimientos del producto.
Salidas
Definición y establecimiento de una línea base de requerimientos. Matriz de viabilidad de requisitos. Plan de desarrollo software. Plan de diseño a alto nivel.
Controles
La revisión de los requerimientos del software. La revisión del diseño preliminarmente. La verificación de que se han llevado a cabo las salidas necesarias. La comprobación de que el contenido de las salidas es correcto, consistente y completo.

Tabla 17. Elementos y actividades de la Fase 3 del SQA.

4. Diseño del aseguramiento: La fase de diseño del aseguramiento ejecuta los controles adecuados para asegurar que el diseño se ha realizado de acuerdo con las políticas y los estándares establecidos. Además, en ella se controla que las salidas necesarias han sido completadas.

Entradas
Definición y establecimiento de una línea base de requerimientos. Matriz de viabilidad de requisitos. Plan de desarrollo software. Plan de diseño a alto nivel.
Salidas
Plan de diseño detallado y procesos de diseño. Definición de métricas para su uso en los procesos: Métricas de producto, métricas de proceso y métricas de calidad.
Controles
La realización de un tutorial de diseño y de inspección. La evaluación del proceso de diseño. La verificación de que se han llevado a cabo las salidas necesarias. La comprobación de que el contenido de las salidas es correcto, consistente y completo.

Tabla 18. Elementos y actividades de la Fase 4 del SQA.

5. Aseguramiento del desarrollo: La fase de aseguramiento del desarrollo garantiza que el equipo de desarrollo sigue los procesos y los estándares de codificación establecidos.

Entradas
Manual de procedimientos y estándares. Plan de desarrollo software. Plan de diseño detallado.
Salidas
Código (producto software para probar) Plan y casos de prueba. Métricas.
Controles
La verificación de pruebas unitarias. La realización de auditorías aleatorias a través de tutoriales, inspecciones revisiones de pares del código. La planificación de pruebas de auditoría. La comprobación de que el código se ajuste a los estándares. La verificación de que se han llevado a cabo las salidas necesarias. La comprobación de que el contenido de las salidas es correcto, consistente y completo.

Tabla 19. Elementos y actividades de la Fase 5 del SQA.

6. Aseguramiento de las pruebas: La fase de aseguramiento de las pruebas ofrece las garantías suficientes de que las pruebas se han realizado y de que los defectos se han seguido y registrado.

Entradas
Plan y casos de prueba.
Salidas
Informe del sumario de pruebas. Plan de implementación. Métricas.
Controles
La revisión del informe del sumario de pruebas. La revisión de los procedimientos de pruebas: Datos de prueba, entorno de prueba, y pruebas automatizadas. La revisión de plan de implementación. La verificación de que se han llevado a cabo las salidas necesarias. La comprobación de que el contenido de las salidas es correcto, consistente y completo.

Tabla 20. Elementos y actividades de la Fase 6 del SQA.

7. Aseguramiento de la implementación: El aseguramiento de la implementación ofrece suficientes garantías de que los pasos de la implementación se han completado.

Entradas
Plan de implementación.
Salidas
Software implementado.
Controles
La revisión de los productos preparados. La verificación de que se han llevado a cabo las salidas necesarias. La comprobación de que el contenido de las salidas es correcto, consistente y completo.

Tabla 21. Elementos y actividades de la Fase 7 del SQA.

8. Fin del SQA: En el fin del ciclo de SQA se completan todos los procesos de aseguramiento, finalizando las actividades que ya han sido completadas.

Entradas
Métricas. Documentación de las fases anteriores.
Salidas
Informe del sumario de SQA. Publicar las revisiones del proyecto. Métricas
Controles
La verificación de que se han llevado a cabo las salidas necesarias. La comprobación de que el contenido de las salidas es correcto, consistente y completo.

Tabla 22. Elementos y actividades de la Fase 8 del SQA.

3.2.7.1.3 El SQA y la verificación y validación

Los procesos de ingeniería del software emplean multitud de procesos de soporte [WALLACE06]. Estos son empleados con el fin de alcanzar sus objetivos y llevar a cabo productos de calidad. Dos de ellos, estrechamente relacionados con la calidad del software y a menudo combinados entre sí, son el aseguramiento de la calidad del software (SQA) y la validación y verificación (V&V). Ambos añaden un valor de calidad y localizan posibles problemas del software, pero difieren en cómo se llevan a cabo.

Tanto el SQA como la V&V permiten gestionar el alcance de la calidad de los productos en cada etapa del desarrollo o del mantenimiento. El alcance de estos, se obtiene a través de las medidas y de los datos producidos por las tareas a lo largo del ciclo de vida del desarrollo.

El proceso de SQA [WALLACE06] dota de una determinada garantía de calidad a los productos y a los procesos software a lo largo de sus respectivos ciclos de vida. Esto es debido a que se produce durante la especificación de requisitos de las actividades, en su planificación. De tal forma, que si un problema es clara y adecuadamente tratado, su solución debe ser adecuadamente definida y expresada en los requerimientos. El SQA busca conservar la calidad a través del desarrollo y el mantenimiento del producto. Para ello, ejecuta un conjunto de actividades en cada área, las cuales proporcionan una temprana identificación de problemas. Mientras, los diferentes roles de SQA en los procesos tratan de asegurar que su planificación resulte adecuada, y su posterior implementación sea llevada a cabo.

Por el contrario, el proceso de V&V [WALLACE06] determina si los productos de una actividad de desarrollo o de mantenimiento se encuentran conforme a las necesidades de la propia actividad y a la de sus antecesoras, y si el producto software final satisface su propósito de uso y las necesidades del usuario. Por ello, se dice que la verificación busca garantizar que los productos sean construidos de forma correcta, en el sentido de que los productos de una actividad cumplan los requerimientos impuestos en actividades previas. Por otra parte, la validación también busca garantizar que los productos sean construidos de forma correcta, pero atendiendo a que respeten sus usos específicos. Ambos procesos comienzan en fases tempranas del desarrollo o del mantenimiento, ofreciendo una valoración de cada producto relativo, tanto a su inmediato predecesor como al sistema de requerimientos que debe satisfacer.

SQA	V&V
Procesos y actividades.	Productos.
Planificación de requisitos.	Satisfacción propósito de uso.
Planificación y aplicación de actividades.	Satisfacción necesidades de usuario.
Identificación de problemas.	Cumplimiento de requerimientos.
Comienzo en fases tempranas.	Comienzo en fases tempranas.
Control y gestión de cambios	

Tabla 23. Características de SQA y V&V.

Es importante destacar que al igual que existen diferencias entre ambos procesos, es preferible que el personal dedicado a cada uno sea independiente. Ya desde 1986 en la publicación “*A SURVEY OF SOFTWARE QUALITY ASSURANCE*” para el DoD, se viene reflejando como más del 53% de las empresas que trabajan para el propio DoD deciden separar las labores de ambos grupos. Aunque no hay que olvidar que ambos procesos están destinados a favorecer la calidad del software y tal y como se ha mencionado con anterioridad es importante la colaboración entre ambos grupos.

3.2.7.2 El grupo SQA

Los *grupos de aseguramiento de la calidad software (SQA)* se definen como los conjuntos de personas, con la formación y las habilidades necesarias, que aseguran la toma de acciones oportunas durante el proceso de desarrollo software, de tal forma que el producto cumpla con los requerimientos técnicos previamente establecidos.

Dichos grupos, pueden estar formados por una sola una persona trabajando a tiempo parcial o un grupo de ingenieros de calidad [BUMS03]. Suelen ser solicitados en los desarrollos de alto riesgo y en los sistemas de misión crítica, sin embargo, siempre es beneficioso contar con controles de calidad en niveles altos de implementación [BOGER84].

El grupo de SQA posee la responsabilidad de planificar, supervisar, registrar, analizar y reportar el aseguramiento de la calidad. Todas estas actividades son llevadas a cabo mediante el uso de distintas técnicas y herramientas de control y calidad como son [GILL05]:

- Los métodos y las herramientas de análisis, de diseño, de codificación y de las prueba.
- Las revisiones de las técnicas formales aplicadas durante cada paso de la ingeniería del software.
- Las pruebas del software y del diseño de casos de prueba.
- El control de la documentación y de los cambios realizados.
- Los procedimientos que aseguren un ajuste a los estándares de desarrollo del equipo lógico.
- Los mecanismos de medida y de información.
- Las métricas para medir la calidad software.

El grupo de SQA ofrece un punto de vista independiente del de la dirección del proyecto y del desarrollo. Actuando como la voz del cliente y asegurando de esta forma que la calidad sea cuidadosamente considerada en cada paso del desarrollo [OREGAN02].

En esta organización, se informa directamente al gerente por encima de otros niveles dentro del proyecto. El grupo debe establecer la forma de comunicarse entre ellos mismos, el equipo de desarrollo y el equipo de pruebas [BUMS03]. Así, si un grupo SQA identifica problemas que influyen en la calidad del producto, informa a la dirección, y ésta debe tratar los problemas y asignar recursos para solucionarlos.

La independencia que posee este grupo, les permite además establecer normas y directrices, medir los resultados y, aprobar los procesos que abarquen múltiples proyectos [ANDREWS04]. Aunque también se encarga de supervisar la labor del personal, para lo cual es importante que trabaje conjuntamente al equipo de desarrollo. En busca de la eficacia, se debe encontrar la manera de poder trabajar con todos los proyectos que se tratan y de moderar el entusiasmo por la calidad, con el fin de que los proyectos software sean viables y puedan ser comercializados [AMESC02].

Las actividades de un grupo SQA normalmente incluyen actividades de prueba del software, para verificar su corrección, y también auditorías de calidad de los distintos grupos involucrados en desarrollo del mismo [OREGAN02]. Para dar apoyo a estas cuestiones, el grupo SQA trabaja con los gerentes del proyecto y los probadores para desarrollar políticas y planes del aseguramiento de la calidad para cada proyecto. Por lo cual, se involucra en labores de recogida, de análisis, de registro y de informe de datos como: las revisiones y las auditorías, el registro y el seguimiento de problemas, y la verificación de las correcciones que se han llevado a cabo [BUMS03]. Estas actividades se evalúan mediante su comparación con el plan, las normas y los procedimientos de desarrollo software establecidos [AMESC02].

Ventajas
La independencia directiva. La constancia desde las primeras fases del ciclo de vida. La objetividad y la eficiencia.
Inconvenientes
El alto coste, factor clave que limita su implantación. La posibilidad de favorecer la dejadez de los desarrolladores.

Tabla 24. Ventajas e inconvenientes de los grupos SQA.

3.2.8 Outsourcing

Nacida en Estados Unidos en 1989, la **internalización u outsourcing**, traslada funciones de una organización a proveedores externos. Funciones como la información, que dada su cada vez mayor importancia en la sociedad actual se la llega a considerar como un recurso estratégico. Para una gestión eficiente de la misma, son necesarias tecnologías de la información (T.I.) y sistemas de información (S.I.) que sirvan de soporte. El uso dichas tecnologías, cada vez más sofisticadas, supone que no todas las empresas se encuentren en disposición de acceder a ellas, e incluso aunque lo estén, que les resulte preferible externalizarlas que asumirlas como propias [SUESC00]. Lo cual, ha llevado a que en la última década el número de empresas que han optado por externalizar alguna de las funciones de su S.I. se ha incrementado exponencialmente. Si bien en décadas anteriores las empresas externalizaban las operaciones de carácter rutinario y

repetitivo, en la década de los noventa se produce un cambio en la tendencia. De esta forma se impulsa la competitividad de la empresa en un mundo de rápido desarrollo. Las empresas contratan a proveedores que ayudan a su organización a ofrecer mejores servicios y a adquirir y mantener una ventaja competitiva. Dado lo cual, se comienza a concebir el outsourcing como una opción estratégica, donde es posible externalizar la mayor parte del S.I., como se puso de manifiesto en uno de los casos pioneros y posiblemente el más conocido de ellos, el de Eastman Kodak [SUESC00].

El éxito conseguido por *Kodak Inc.* (efecto Kodak) debido al mega-contrato firmado con *IBM*, *Digital Equipment Corp.* y *Businessland* para externalizar la mayor parte de sus sistemas de información (TI Outsourcing) [SAIZ], llevó a una nueva concepción del outsourcing como estrategia. Tras lo cual, se comenzó a utilizar como una opción reductora de costes y de creación de holding empresariales (crecimiento en red) por empresas transnacionales situadas a lo largo todo el mundo [SAIZ].

Durante su desarrollo, países como India, Irlanda, e Israel han establecido una gran presencia en este mercado [CORRIV07]. Por ejemplo, *Microsoft*, *SAP* y otras empresas poseen centros de desarrollo en los Estados Unidos, Europa, Israel y la India; *IBM*, *Hewlett Packard* y *Electronic Data Systems* están desarrollando software y externalizando soporte a México, India, y Brasil; *BearingPoint*, antes *KPMG Consulting*, ofrece servicio de desarrollo software desde China. Pero además existen muchos otros ejemplos de centros de desarrollo con sede en Irlanda, el Reino Unido, Singapur, Canadá, Rusia, y en diferentes localizaciones [SIKKA].

Con la evolución del contexto empresarial, volviéndose este cada vez más dinámico y competitivo, las empresas se han centrado en sus principales áreas de negocio como una manera de obtener una ventaja competitiva [HAMEL90]. Así, las organizaciones que externalizan servicios se centran en las competencias que agregan un valor único a sus clientes y subcontrata actividades secundarias o no críticas [QUINN94].

La aplicación del outsourcing en el ámbito de las TI ha tenido un gran impacto gracias al desarrollo de las comunicaciones, facilitando el desarrollo de productos a grandes distancias entre el suministrador y el cliente [CAPAC]. Con el fin de poder desarrollar estos productos distribuidos de la manera más eficaz, las compañías necesitan trabajadores altamente cualificados y utilizando a menudo modelos de madurez de procesos como CMM, CMMI e ISO 900x en el desarrollo [FERG04].

3.2.8.1 El concepto del outsourcing

El **outsourcing** se trata de la decisión de la organización de trasladar parte, o bien, todas las funciones de su organización a proveedores externos de servicio, de manera que pueda cumplir sus objetivos satisfactoriamente [DIBB04]. Fue aplicado como un intento de reducir sus costes fijos y, de forma simultánea, flexibilizar su estructura productiva y optimizar el uso de recursos. A menudo las empresas delegan actividades, que no forma parte de sus habilidades principales, a un tercero, que se especializa en esa operación. Por habilidades principales se entienden todas aquellas actividades que conforman el negocio central de la empresa y sobre las cuales se posee ventaja competitiva con respecto a la competencia.

Por lo tanto, el outsourcing es el modo de realizar tareas que no están relacionadas con el corazón del negocio de forma más eficiente y a un menor costo, sin necesidad de utilizar recursos que puedan ser fundamentales para la actividad principal, así permite a las compañías enfocarse en lo que mejor hacen [SAIZ]. De esta forma, los proveedores pueden especializarse en sus procesos o funciones de negocios particulares, transmitiendo las ventajas de esa especialización a sus propios clientes, mientras que la organización que contrata outsourcing puede, a su vez, enfocar el esfuerzo en sus competencias primarias.

A la hora de establecer la externalización de un proceso o un servicio existen diferentes aspectos que pueden motivar que una organización tome una determinada decisión, entre ellos se distinguen los siguientes [MELLON02] [DIBB04]:

- **Económicos**, se trata de la principal motivación, una empresa elige externalizar todo o parte de su sistema de información (S.I), con el fin de reducir costes o mejorar su rendimiento.
- **Estratégico**, normalmente al optar por el outsourcing la organización centra sus esfuerzos en desarrollar sus actividades principales o añade valor competitivo en actividades donde no destacaba.
- **Tecnológicos**, su motivación se basa en la actualización de los recursos y las destrezas relacionadas con las tecnologías que la empresa ya tiene o la introducción de nuevos.
- **Políticas**, son motivados por las dificultades que la organización tiene para gestionar su sistema de información.

No obstante, no se debe olvidar antes de externalizar una función estudiar las características del futuro proveedor. Estos suelen estar representados en función de la distancia que existe entre el cliente y ellos mismos.

A continuación, se muestra una tabla resumen con las principales ventajas y desventajas que presenta la aplicación de una organización outsourcing en una organización:

Ventajas
La posibilidad de acceder a expertos y a tecnologías de punta. La posibilidad de transformar costos fijos en costos variables. La flexibilidad a la hora de tomar decisiones. La posibilidad de reenfocar las estrategias de la compañía en su negocio principal.
Inconvenientes
La posibilidad de dar a conocer aspectos íntimos de la organización. La falta de compromiso. La existencia de un menor control. La posible pérdida de personal calificado y con experiencia La excesiva dependencia de los proveedores. La empresa abandona una actividad.

Tabla 25. Características del outsourcing.

3.2.8.2 El ciclo de vida

Con el fin de establecer un proceso de outsourcing óptimo la *Asociación Internacional de Profesionales Outsourcing (IAOP)* define nueve etapas [IAOP] que guían su desarrollo, tal y como se refleja en la siguiente figura:

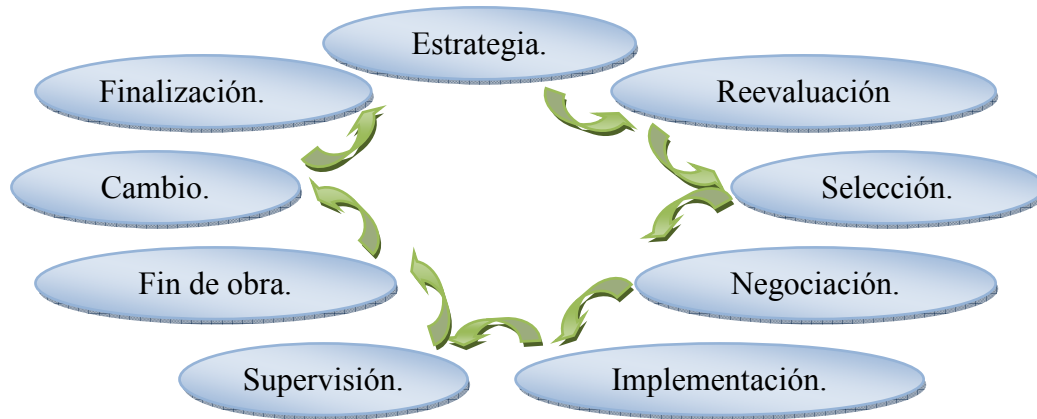


Figura 19. Ciclo de vida del outsourcing.

Definición de la estrategia: El outsourcing es una decisión estratégica desarrollada normalmente en niveles maduros de una empresa. Tal y como se ha definido con anterioridad, la externalización puede actuar como un diferenciador clave, que proporciona a un negocio una ventaja competitiva sobre sus competidores. Por tanto, establecer una estrategia correcta cobra especial importancia.

Reevaluación: La reevaluación no siempre es necesaria, pero es importante evaluar los procesos actuales de una organización y estudiar los posibles beneficios que podrían obtenerse con su externalización.

Selección: La fase de selección, tal vez la fase más importante, trata de evaluar las diferentes ofertas de proveedores y establecer a uno de ellos.

Negociación: Durante esta fase se procede a la negociación de los contratos, los horarios y los acuerdos anexos a la firma del contrato definitivo, la cual suele estar precedida por una profunda revisión del proceso por ambas partes.

Aplicación: La fase de aplicación consiste en la puesta en marcha de las actividades de planificación y de aplicación del contrato, así como en establecer el presupuesto detallado y las funciones administrativas necesarias para su gestión.

Supervisión de la gestión: La fase de supervisión de la gestión abarca todas las actividades necesarias para gestionar el programa, y lograr los resultados contratados. Para lo cual, incluye la supervisión del rendimiento, de la administración del contrato, de los acuerdos, de la integración y de la entrega.

Fin de obra: La fase de fin de obra abarca todas las actividades de finalización presentes en la fase de construcción, incluyendo cualquier programa de aceptación y la introducción de nuevos servicios.

Cambio: La fase de control debe estudiar los cambios producidos durante el desarrollo del outsourcing. Todo contrato debe estar sujeto a una posibilidad de cambio durante el desarrollo del mismo, con intención de minimizar su impacto sobre el producto final.

Finalización: La fase de finalización hace frente al fin del outsourcing, ya sea por el flujo del tiempo, de mutuo acuerdo o, a veces por problemas. Debiendo permitir, en la medida de lo posible, el cambio ordenado, la asistencia posterior a la salida, la transferencia del personal y de la propiedad de los datos e intelectual.

3.2.8.3 Tipos de acuerdos outsourcing

Existen numerosos planteamientos *outsourcing*, entre los que se destaca el de Millar, el cual define cuatro tipos básicos de acuerdos de outsourcing, en los que se refleja el tipo de relación comercial entre el cliente y el proveedor [DIBB04]:

- **Outsourcing general:** El outsourcing general se encuentra representado por tres alternativas:
 - *Outsourcing selectivo*, cuando se concede la elaboración de un área específica de una actividad de IS a una tercera parte.
 - *Outsourcing de valor añadido*, cuando se delega a una tercera parte algún área de la actividad de IS con la intención de que se encargue de dar valor añadido a la actividad, ya que la propia empresa no lo es capaz de ofrecer.
 - *Outsourcing cooperativo*, cuando un objetivo es desarrollado conjuntamente por la organización y a una organización externa.
- **Outsourcing de transición:** El outsourcing de transición engloba la migración de una plataforma tecnológica. Se entienden tres tipos diferentes de transición:
 - *Gestión de sistemas legales.*
 - *Transición a nuevas tecnologías o sistemas.*
 - *Establecer y gestionar una nueva plataforma.*
- **Outsourcing de un proceso de negocio:** El outsourcing de un proceso de negocio se refiere a la relación de outsourcing donde una tercera parte proveedora es responsable de desarrollar una función de negocios entera para la organización.
- **Beneficios de los negocios de los contratantes:** El outsourcing de beneficios de los negocios de los contratantes se refiere a un acuerdo contractual que define la contribución del proveedor al cliente en términos de beneficios específicos para la empresa, y define el pago que el cliente hará basándose en la capacidad del proveedor para entregar los beneficios. El objetivo es hacer coincidir los costes reales con reales los beneficios y compartir los riesgos.

3.2.8.4 Modelos de provisión

En el desarrollo del outsourcing además de la ejecución de una estrategia bien definida, es importante la construcción y la gestión de asociaciones eficaces. Para lo cual, se ha de aplicar la mejor combinación de comunicaciones e infraestructuras posibles, con el fin de mantener la competitividad con las prácticas locales [SIKKA].

Así, una vez externalizado un servicio, su implicación con la empresa se diferencia en función de la distribución de sus proveedores. La distancia existente entre los clientes de outsourcing y sus suministradores ha producido una clasificación de los mismos, tal y como se muestran en la siguiente figura [GASCO06].

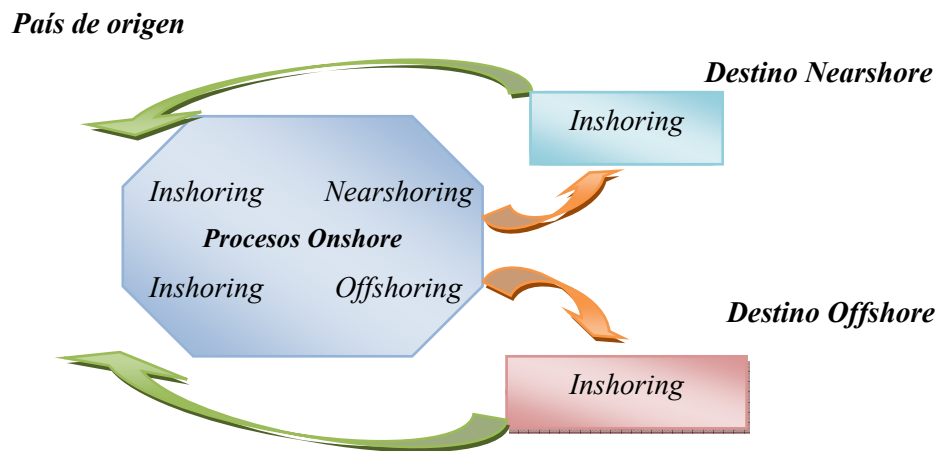


Figura 20. Relación entre los tipos de outsourcing.

En la Figura 20 se utiliza el término *inshoring* para referirse a la reubicación de los procesos de negocio de costes altos en los países de costes bajos, desde el punto de vista del país de destino [ERBER05]. Además, se pueden observar los distintos tipos de enfoques outsourcing utilizados en la actualidad. Estos comparten con el modelo básico sus características (Tabla 25), diferenciándose normalmente en términos de *costes* y *diferencias culturales en función de la distancia* (Tabla 26). Tanto las ventajas como las desventajas existentes son directamente *proporcionales* a la distancia existente entre el cliente y el suministrador [BOEHM81].

Siguiendo la línea de algunos estudios [81, 88], se promueve la idea de dividir equipos geográficamente, obteniendo grupos offshore y onshore. Favoreciendo de este modo, el contacto con el cliente principalmente en funciones de análisis y diseño, sin perder el ahorro económico que supone el offshore. Por contra, en distintas investigaciones [82, 83, 84] se muestra el aumento del tiempo de trabajo en equipos distribuidos a causa de las dificultades de comunicación, y la necesidad de disminuir los tiempos en la comunicación. A consecuencia de lo cual se aconseja minimizar el acoplamiento de los equipos involucrados, dividiendo los subsistemas por zonas de trabajo, o las fases del proceso por zonas del proceso [85, 86, 87].

En la actualidad, existen distintas estrategias para proveer servicios de outsourcing, a continuación se establecen una serie de modelos [SOOD05], los cuales pueden sufrir variaciones con el fin de acomodarse a la naturaleza del proyecto.

Ventajas
El ahorro de costes. La ampliación de nuevos mercados. La búsqueda de mercados más eficientes.
Inconvenientes
Los costes ocultos. La pobreza de infraestructuras. Las diferencias culturales.

Tabla 26. Ventajas y riesgos en función de la distancia del outsourcing.

Modelo Onsite.

El *onshore u onsite* se encuentra orientado a la distribución de los recursos del proceso de calidad contratados en el propio cliente, eliminando las barreras entre el cliente y el proveedor, facilitando así la familiarización con el desarrollo por parte del grupo de calidad [TINN01].

Este tipo de outsourcing facilita las comunicaciones, además familiariza a los recursos con el proyecto aunque precisamente es donde aumenta el coste frente al nearshore y al offshore.

Resulta frecuente utilizar un equipo onsite en comunicación constante con un equipo offshore. El caso más simple del modelo onsite es el aumento de recursos o de provisión temporal de personal. En este escenario, el personal del cliente será responsable de la dirección de los proveedores, lo que implica cierta sobrecarga.

Ventajas
La flexibilidad de los recursos. El control continuo del progreso. La posibilidad de tomar decisiones y de cambiar el rumbo con rapidez. La minimización de la necesidad de largas y costosas comunicaciones.
Inconvenientes
La posibilidad de producirse demasiados cambios en la dirección. La sobrecarga en costos de infraestructura y gestión del proyecto. La gestión del proyecto podría saturarse con la adición de nuevos proveedores.

Tabla 27. Ventajas y riesgos en función del modelo onsite.

Modelo Offsite.

El *modelo offsite* [SOOD05] se encuentra planteado desde el punto de vista que el proveedor tiene una oficina cercana al cliente (en la misma ciudad). De esta forma, los costos de oficinas, computadoras y materiales, corren por cuenta del proveedor. Aun así, el cliente paga por esto de una u otra forma, por lo que se deberá evaluar si el proveedor

ofrece mejores costos que los propios. Sin embargo, la provisión de una red segura estará a cargo del cliente.

Es un modelo más restrictivo que el onsite, y necesita cierta estructura. Aquí el gerente de proyectos del cliente se relaciona con el coordinador offsite (o gerente de proyecto offsite), quien es el encargado de administrar los recursos. El equipo offsite visita al onsite, muy a menudo al inicio, para establecer alcance y entregables. Pero a medida que el proyecto avance, sólo se realizarán reuniones programadas para reportar avance.

Ventajas
La flexibilidad de los recursos
El uso de una metodología, ya que los equipos trabajan separados.
La proximidad física reduce los costos de comunicación.
Una menor sobrecarga de costos que en el modelo onsite.
La gerencia mantiene un enfoque de comunicación más estructurado.
Inconvenientes
La pérdida de productividad debido a los viajes entre oficinas.

Tabla 28. Ventajas y riesgos en función del modelo offsite.

Modelo Offshore.

El *modelo offshore* se encuentra basado en que toda la actividad se realice en otro país. El personal del cliente debe tratar directamente con el equipo offshore. En lugar de tener un contacto cara a cara diario, se debe programar un conjunto de reuniones pre-programadas. Así, el término de *offshore* puede definirse como el traslado de procesos de negocio (incluyendo la producción, distribución y servicios empresariales, así como actividades básicas, como la investigación y desarrollo) a lugares de menor costo fuera de las fronteras nacionales, asumiendo la perspectiva del país de origen [ERBER05].

Las actividades típicas de offshore se basan en los S.I, siendo las labores de programación uno de los servicios más demandados. Aunque también los servicios de calidad son otro de los incentivos para la subcontratación. El socio de outsourcing ideal debe de asegurar un trabajo de calidad a precios bajos y una moderna infraestructura de TI, garantizando el cumplimiento de las normas de calidad internacional [MELLON02].

El modelo offshore puede funcionar muy bien cuando el alcance de las entregas se encuentra claramente definido y no existen cambios en la gestión del proyecto. Este modelo puede ser desplegado exitosamente en tareas que requieren personal las 24 horas (24x7). En modo similar al modelo offsite, el gerente de proyecto del cliente, usa al coordinador offshore para administrar los recursos de la locación offshore. Entre estos equipos la comunican se produce a través del email, herramientas de colaboración online, tele y video conferencias.

Ventajas
La reducción de costos.
La alta flexibilidad de los recursos.
Los recursos disponibles offshore son jóvenes y motivados.
El proveedor garantiza una fuerza laboral de alta calidad.

Ventajas
La posibilidad de operar 24x7. La mayoría del sobre-coste está incluido en las bajas tasas que se pagan. La posibilidad de manejar proyectos basados en costes y tiempo fijo. La liberación del equipo onsite.
Inconvenientes
El equipo offshore seguramente será muy joven y sin experiencia. La necesidad de excesiva documentación. El incremento del riesgo de largas y costosas comunicaciones. La visibilidad del trabajo no tiene una base día a día. Las diferencias culturales pueden causar problemas de comunicación.

Tabla 29. Ventajas y riesgos en función del modelo offshore.

Modelo Híbrido

El **modelo híbrido** [SOOD05] es una variante de los modelos onsite y offshore puros, para permitir que la mayor carga de trabajo se realice de forma offshore. En este modelo, el equipo onsite es administrado por el proveedor, quien es responsable de la transferencia del conocimiento, así como de la integración y ejecución de las tareas. El equipo onsite trabaja junto al cliente para empaquetar y enviar el trabajo a implementar al equipo offshore. El gerente de proyecto del cliente delega en el proveedor el control y la coordinación de todas las actividades incluyendo los equipos onsite y offshore.

Este modelo es el más exitoso y más ampliamente usado, con pequeñas variaciones dependiendo del trabajo a realizar [SOOD05]. Posee todas las ventajas del modelo offshore combinadas con las del modelo onsite.

Ventajas
La superposición de zonas horarias no es un problema (para el cliente), ya que el equipo onsite es responsable de la comunicación con el equipo offshore. Las diferencias culturales son subsanadas, ya que la comunicación se realiza usando canales definidos. La existencia de una mayor visibilidad, dado que el equipo onsite muestra el progreso.
Inconvenientes
La necesidad de una estructura de equipo compleja. La existencia de múltiples niveles de comunicación.

Tabla 30. Ventajas y riesgos en función del modelo híbrido.

Modelo Nearshore.

La utilización del **modelo nearshore** está asociada con la mayor proximidad del outsourcing respecto al *offshore*. Presenta una vía para competir con India, principal sede *offshore*, ofreciendo una alternativa a su reducción de costes y abriendo nuevos mercados [TINN01].

Su principal ventaja sobre el modelo offshore es que el tiempo de viaje se reduce a un par de horas [SOOD05]. Por lo cual, comparte con el modelo offshore ventajas y desventajas, aunque ambas reducidas en función de su mayor proximidad.

3.2.8.5 El estado del outsourcing en la actualidad

A pesar de los problemas existentes en la economía global, tal y como se puede observar en el Figura 21, donde casi 2/3 de las organizaciones implicadas en el estudio propuesto por la Asociación Internacional de Profesionales de Outsourcing (IAOP) [CORB09] informan de un rendimiento negativo en general en el periodo de crisis, o probablemente debido a ello, uno de los pocos sectores que ha seguido creciendo es el outsourcing [SCOTT09].

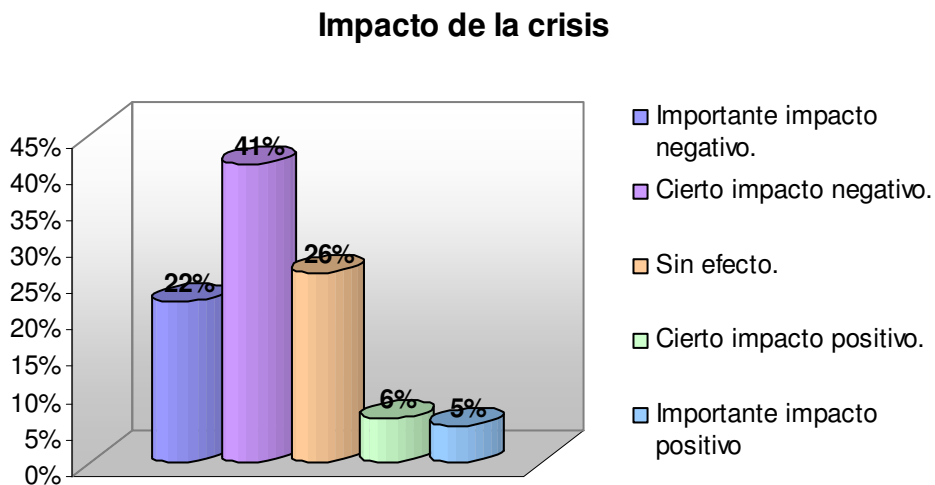


Figura 21. Impacto de la crisis (miembros de la IAOP) [CORB09].

Las empresas que buscan la reducción de gastos ante la bajada de la demanda están más dispuestas que nunca a operar a través de proveedores de bajo costo. Esto se observa en la tendencia a la bajada de precios y volúmenes reflejada en Figura 22 y en la subida de importancia de los descuentos y de las flexibilidad de los contratos outsourcing tal y como refleja la IAOP. Como resultado, los productos ofertados en base a soporte y servicios de TI vio crecer los ingresos del 2008 [SCOTT09] en un promedio de 13% y ampliar su flujo de trabajo en un 18%.

Estado actual del outsourcing.

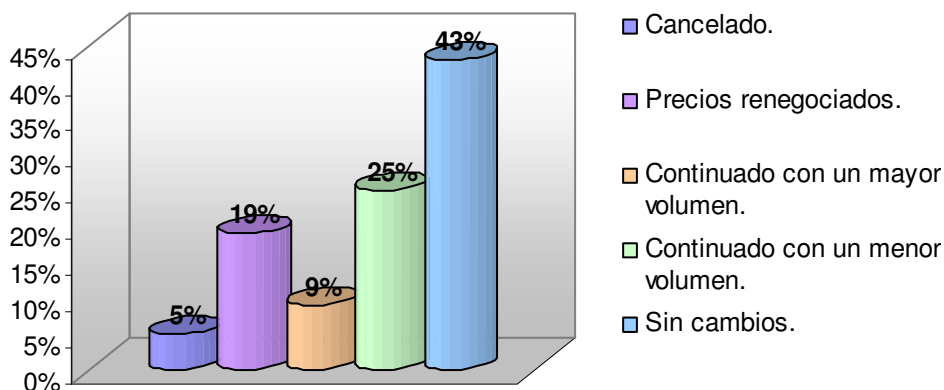


Figura 22. Estado actual del outsourcing (miembros de la IAOP) [CORB09].

Casos de negocio solicitados.

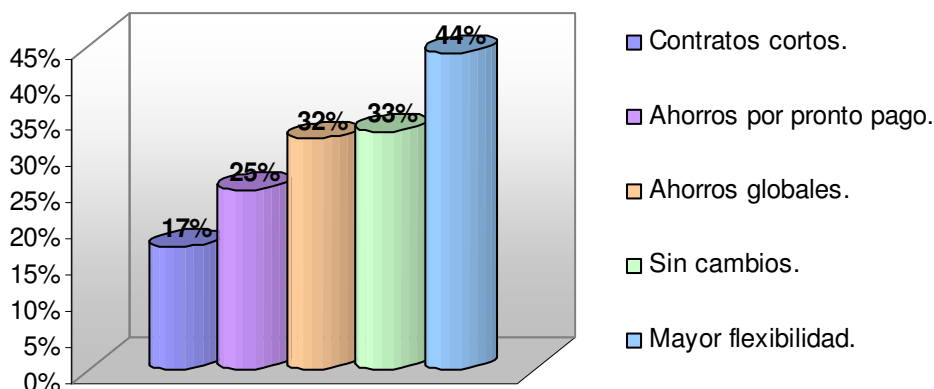


Figura 23. Casos de negocios solicitados miembros de la IAOP [CORB09].

Por supuesto, esto no ha supuesto la disminución de la tasa de despidos en los países desarrollados, pero sí que los principales proveedores de outsourcing, desde IBM (IBM) y Capgemini (CAPP.PA) a los gigantes indios como Wipro (WIT) e Infosys (INFY), sobrevivan mejor a la recesión que muchas empresas [SCOTT09].

La situación económica está modificando la subcontratación en sí [SCOTT09]. Los despidos generalizados en el Oeste, especialmente en EE.UU., han inundado el mercado con trabajadores altamente cualificados, ansiosos de puestos de trabajo y dispuestos a aceptar salarios más bajos. Permitiendo, de forma local, ofrecer servicios por un coste mucho menor de lo que se podía hacer hace dos años. Ésta situación, junto al aumento de los costes en los destinos tradicionales de outsourcing en la India y China, como Bangalore y Mumbai, hace que por ejemplo en las ciudades de segundo nivel de EE.UU., como Indianápolis y Boise, el precio sea relativamente competitivo respecto a años anteriores [BROWN09].

Debido a que el envío de trabajo al extranjero ya resultó polémico cuando los tiempos eran buenos, ahora, ante la generalización del desempleo, tiene el potencial de ser alarmante. Por ello, las compañías de carácter local sin pasado outsourcing, no quieren experimentar con el offshore. Así, al igual que pasa con el caso de Indianápolis en EE.UU., en Europa, se buscan ciudades como Gdansk en Polonia y la capital búlgara, Sofia, donde ofrecer soporte a sus TI y a sus procesos de fabricación buscando una mayor seguridad y ahorro (Figura 24), como ha pasado en el caso de Sony (SNE) y Lufthansa (LHAG.DE) [BROWN09].

Selección de proveedores.

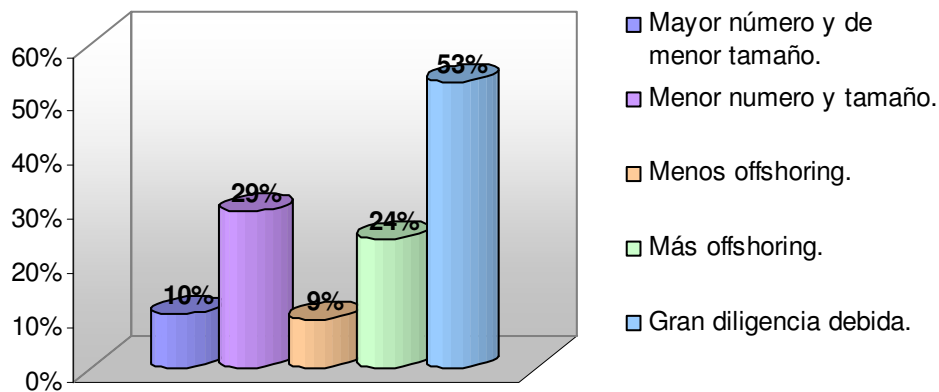


Figura 24. Selección de proveedores miembros de la IAOP [CORB09].

El efecto total se traduce en un aumento de las opciones de localización del outsourcing y del aumento de un mayor número de contratos a un menos coste. Por lo que aunque en 2008 las ofertas cayeron un promedio del 5% al 10% en relación a sus valores anteriores [SCOTT09], el 74% de las organizaciones mantiene la planificación o incorpora mayor perspectiva de outsourcing como respuesta a la crisis.

Estado de las oportunidades de outsourcing.

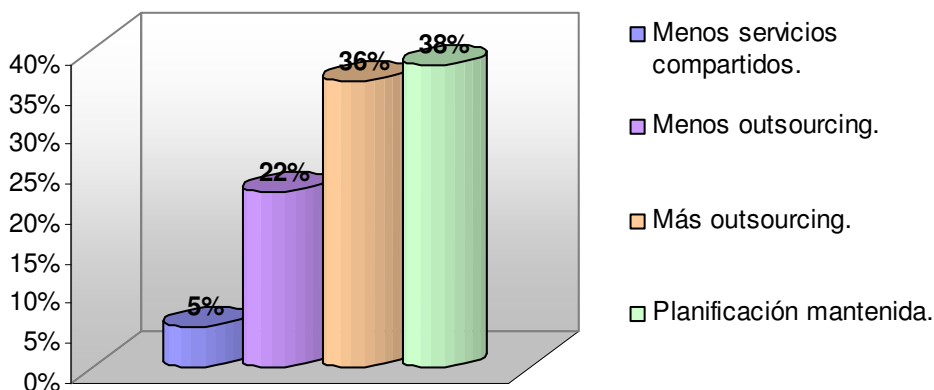


Figura 25. Estado de las oportunidades de outsourcing (miembros de la IAOP).

3.2.8.6 El outsourcing de las pruebas

En la actualidad, la tendencia a optimizar la relación entre las TI y los negocios, se basa en la externalización de aquellas tareas que no resultan clave, y que pueden ser prestadas por terceros. Lo cual, facilita una mayor especialización de los perfiles de la compañía en la gestión de servicios de TI. Esta tendencia ha provocado un incremento de la externalización de los proyectos de desarrollo software, así como del alcance y de la profundidad de los procesos de aseguramiento de la calidad de los mismos [QUILES07].

El outsourcing como extensión de la gestión del proyecto y de las herramientas de prueba, cada vez ofrece menos dificultades en la gestión de actividades de equipos divididos en distintas áreas geográficas. A su vez, la gestión del código y de los sistemas de seguimiento de fallos se encuentra cada vez más globalizada, y su mantenimiento y su actualización se realizan en tiempo real. [SIKKA].

En el desarrollo de una organización de pruebas outsourcing, todo o al menos alguna parte del proceso de pruebas se realiza por una tercera parte, es decir, desde estrategias de pruebas, planificación, diseño, ejecución, y/o automatización. Para ello, es muy importante que se produzca bajo un marco bien definido, contando con buenos contratos y altos niveles en estándares de calidad [CRAIG02].

La realización de pruebas independientes, implicadas directamente en la calidad del producto, ofrece beneficios desde el punto de vista de la obtención de objetivos por parte de las organizaciones involucradas. Para la organización que realiza las pruebas, el objetivo final es el de encontrar defectos, mientras que para la organización que desarrolla el producto el objetivo es el de construirlo. La consecución de estos objetivos en organizaciones de pequeño y mediano porte, provoca que la subcontratación de los servicios de prueba resulte imprescindible para implementar pruebas independientes. Debido a que es difícil sustentar la existencia de un área dedicada exclusivamente a las pruebas dentro de la misma organización.

El outsourcing de pruebas resulta, por lo tanto, una manera de mejorar las carencias de la organización, no de eliminar responsabilidades de la misma. Las empresas dedicadas a este tipo de labores suelen contar con una gran experiencia y tener excelentes herramientas y entornos, pero no tienen una visión clara de la funcionalidad del negocio. Siendo así, necesario la supervisión de este proceso de pruebas [CRAIG02].

Normalmente, cuando se produce el diseño de un sistema complejo, de gran envergadura y de desarrollo global, se descompone en múltiples actividades, procesos y recursos, y esto no es ajeno al SQA. Por ello, se debe determinar una relación de cooperación entre los diferentes equipos que poseen distintas habilidades, provocando una alta necesidad de comunicación y de cooperación entre las diferentes partes del proyecto [PAN06].

A lo largo de este documento se viene reflejando la amplitud del proceso de pruebas, por ello no resulta de extrañar la gran variedad proveedores de pruebas que han ido surgiendo conforme a la evolución de las mismas. A continuación, se establecen los servicios outsourcing más representativos, presentes en algunos de los proveedores de pruebas más importantes. Los resultados obtenidos son propuestos en función de los diferentes tipos de centros de pruebas existentes:

Centros expertos en pruebas (CEP).

El centro de expertos en pruebas se trata de una primera aproximación de las empresas al establecimiento de un proceso de negocio principal centrado en las mismas. En dichos centros, se aúna el personal y las infraestructuras necesarias, en un marco común de estandarización de procesos de calidad y optimización de recursos, con el fin de generar beneficios extraordinarios a la organización. Los CEP cumplen principalmente con una función de suministro y de asesoramiento, estableciendo una "obligación de esfuerzo" en la mayoría de de sus servicios [IEEE98] debido a que sus labores están más cerca de la colaboración que de la resolución de tareas. Dicho lo cual, se diferencian los siguientes servicios dentro de estos centros:

- **Equipo de pruebas:** Estas organizaciones ofrecen servicios y recursos especializados en pruebas software. Abarcando desde personal para la colaboración en el proceso hasta actividades de prueba específicas.
- **Mejora de la externalización:** A través de actividades de consultoría, estas organizaciones facilitan y mejoran los procesos de outsourcing dada la gran variedad de mercado.
- **Automatización de las pruebas:** Estas organizaciones ofrecen metodologías para automatizar los procesos de prueba permitiendo a la empresa centrarse en su objetivo de negocio sin que necesite invertir en formación adicional, personal o infraestructura.

Debido a que estos centros prestan un servicio más puntual o selectivo al que puedan ofrecer el conjunto de las factorías de pruebas, las cuales serán vistas a continuación, establece un mercado en el que se hace necesaria una menor cantidad de recursos e infraestructuras. Dando lugar a la aparición de pequeñas PYMES como: uTest (<http://www.utest.com/>), Ajilon Global Solutions Center (<http://www.aclabs.com>).

Factorías de pruebas (FP).

Las factorías de pruebas son, de un tiempo a esta parte, un concepto que cobra cada vez mayor presencia a la hora de establecer procesos de calidad dentro de una organización. Basadas en el concepto de las *factorías* software, las cuales son entendidas como una línea de productos software que configura herramientas extensibles, procesos y contenido para automatizar el desarrollo y el mantenimiento de variantes de un producto arquetípico mediante la adaptación, ensamblaje y configuración de componentes basados en *frameworks*. Es decir, se centran en el desarrollo de sistemas software promoviendo la reutilización de arquitecturas, componentes software y conocimiento [BENTON08]. Por lo tanto, y basándose en este concepto, las *factorías de pruebas* se establecen como una forma de llevar a cabo servicios de verificación y validación de manera externalizada, mejorando la capacidad de producción, el tiempo de mercado y variabilizando los costes de los proyectos en función de la carga de trabajo.

Este tipo de organización adquiere “obligaciones de resultados” con la empresa a la que presta el servicio a través de los distintos servicios que ofrece [IEEE98], dotando al producto de un valor añadido dada su especificidad en labores de prueba. La variedad de

servicios que manejan este tipo de factorías resulta muy amplia, abarcando la totalidad de los suministrados por un CEP junto a las propias del modelo:

- **Externalización de un servicio de pruebas software:** Dicho servicio externaliza un proceso completo, con el fin de dar soporte a todas las pruebas de software en curso, según las necesidades de una organización o unidad de negocio. Por lo general, incorpora la gestión de pruebas, las pruebas funcionales y los especialistas en servicios de pruebas cuando sea necesario.
- **Externalización del programa de pruebas:** Mediante este servicio se da cobertura a la necesidad de una organización de externalizar todo su proceso de pruebas y el aseguramiento software de las actividades de una iniciativa o de un programa concreto.

Sus clientes son variados: departamentos, proyectos o sistemas que puedan externalizar sus competencias de pruebas a este tipo de organizaciones. Dicha segmentación en el alcance del mercado permite la presencia muy diversos tipos de empresa ofreciendo sus servicios, abarcando desde grandes empresas a PYMES mucho más modestas.

El hecho de ofrecer la externalización por completo de un proceso supone, normalmente, una gran carga de recursos e infraestructuras, cuyas necesidades puede sobrepasar la oferta de una pequeña empresa. No obstante, existen organizaciones como TestAbil, que carecen de una gran envergadura, y que ofertan dichos servicios. Es ahí donde ganan peso las PYMES más modestas, ya que dotan al servicio de una mayor flexibilidad y calidad local que las grandes empresas. Por ello, en los últimos años las factorías PYME lideran las certificaciones de calidad TI, y son un referente de agilidad y productividad para sus clientes locales, mientras revisan sus planes de negocio para incorporar estrategias de internacionalización.

Pero como es evidente, una empresa pequeña no puede afrontar retos como el caso de la reciente adjudicación del suministro único de las pruebas y la implantación de software de Telefónica por parte de MTP. La necesidad de afrontar dichos retos en base a fuerte infraestructura, la cual a menudo hace gala de centros y recursos localizados a grandes distancias, fomenta una corriente por la cual las empresas centran sus esfuerzos en establecer los servicios de factorías de prueba como proceso de principal de negocio, tal es el caso de: ACUTEST (<http://www.acutest.co.uk>), AppLabs (<http://www.applabs.com>), Sogeti(<http://www.sogeti.com/>) o MTP (<http://www.mtp.es>). Aunque, como es evidente, la expansión de un mercado cada vez más en alza como es el de la calidad no ha dejado indiferente a las grandes consultorías, véase el caso Wipro (www.wipro.com), *la cual establece este tipo de servicios*.

A pesar de los recortes presupuestarios que ha provocado la situación de crisis económica, el 71% de las empresas continúa aún invirtiendo en el desarrollo de nuevas aplicaciones destinadas a incidir en la ventaja competitiva del negocio, aunque no para procesos internos de la organización. Esta es la conclusión de un estudio llevado a cabo por Sogeti y Capgemini [COMK] entre más de 30.000 empresas en los cinco continentes y con especial atención a Europa, Estados Unidos y Asia.

El estudio, concluye también que la crisis ha provocado una merma de los recursos de TI dentro de las empresas, lo que ha llevado a un crecimiento notable del trabajo conjunto de los equipos de desarrolladores y de los equipos de pruebas de las aplicaciones, actividades muy separadas hasta ahora. El 72% de las empresas encuestadas apunta que los profesionales más valorados son los que, a la capacidad de realizar pruebas de calidad de software, les suman conocimientos sobre el entorno de desarrollo.

Además, una encuesta realizada por el Instituto Nacional de Estándares y Tecnología (NIST) [AIPMM10] ha obtenido resultados que indican que las empresas americanas están realizando un gasto próximo a los 60 mil millones dólares en solucionar los defectos que surgen una vez comercializado el producto. Este mismo estudio, destaca que la realización de labores de pruebas y de aseguramiento de la calidad por parte de terceros podría haber resultado un factor reductor de costes en casi \$ 20 mil millones. Sin embargo, según un informe de Forrester publicado en 2008, a pesar de los enormes beneficios potenciales de la subcontratación, entonces no resultaban mayoritarias las organizaciones externalizaban sus actividades de control, sólo el 16% de las organizaciones externalizan las pruebas del software, mientras que otro 16% espera llegar a hacerlo en los próximos 12 meses.

A pesar de estos datos, el mercado de la externalización de las pruebas está experimentando en la actualidad un rápido crecimiento, de acuerdo con esta misma investigación de Forrester, los servicios de externalización de pruebas software sufre un crecimiento del 50% anual, o incluso mayor, tanto para el desarrollo como para las pruebas de sus aplicaciones, e independientemente de las metodologías que empleen.

Por otra parte, más de la mitad de las empresas europeas planifica incrementar el uso de los recursos offshore de pruebas el año próximo. En ambos casos, el crecimiento más importante de ambas prácticas es el europeo, muy por encima del crecimiento experimentado por otras áreas geográficas. En el caso del outsourcing, el 65% de las empresas europeas declara utilizar esta práctica para el desarrollo y las pruebas.

Por todo esto no es de extrañar que según la consultora Nelson Hall [MKM], los servicios de pruebas podrían llegar a mover más de 11.200 millones de dólares en todo el mundo en 2013. Según esta firma, las pruebas del software serán uno de los segmentos con mayor crecimiento en el entorno de las TI debido a la necesidad de las empresas de ser muy ágiles para poner en producción nuevas aplicaciones con alto nivel de calidad, factor fundamental para ser competitivas en sus mercados.

Otro factor que está aumentando el valor de los equipos de pruebas de calidad de software dentro de las organizaciones viene determinado por la mayor presión en los proyectos de desarrollo, a los que se exige eficiencia, consistencia, una metodología que asegure la calidad y un mejor uso de las herramientas automáticas de pruebas. Según Sergio Vernis [COMK], Director General de Sogeti España “estas cifras manifiestan una tendencia muy clara de convergencia entre el desarrollo de aplicaciones y las prácticas de calidad. De hecho, los profesionales de la calidad de software tendrán, día a día, un papel más activo en los procesos de desarrollo”.

Según el estudio de Sogeti y Capgemini [COMK] hay dos tendencias claras de inversión: el desarrollo de aplicaciones en entornos “cloud”, mediante los cuales no hay necesidad de conocer la infraestructura detrás de estos, pasa a ser “una nube” donde las

aplicaciones y servicios pueden fácilmente crecer, funcionar rápido, sin conocer los detalles del funcionamiento de esta “nube”. Este concepto fomenta la aplicación de modelos de outsourcing de pruebas. Sin embargo, la adopción del denominado “agile testing”, por el cual se evitan las metodologías tradicionales de desarrollo basándose en los resultados y con una mayor colaboración entre distintos equipos de personas, minimiza riesgos desarrollando software en cortos espacios de tiempo, fomentando la aparición de pequeños grupos de pruebas en el proceso.

En el caso del desarrollo en entornos “cloud”, por ejemplo, la reducción de costes puede llegar hasta el 50%, el tiempo de entrega se recorta en un 33%, y el de puesta en producción de la aplicación en un 14%. Según el estudio, el 75% de las empresas declara haber comenzado a desarrollar aplicaciones en este entorno, y sólo un 8% de las empresas encuestadas dice haber tenido problemas a la hora de descender la aplicación a una infraestructura física.

Por lo que respecta al denominado “agile testing”, el 60% de las empresas declara que usará esta metodología a lo largo del año para entregar aplicaciones. Entre los beneficios que aporta esta opción, cabe destacar una reducción del 37% del tiempo de puesta en producción, un aumento del 26% de la calidad de la aplicación, una reducción del 23% en la utilización de recursos y un ahorro total de costes del 14%.

A pesar de esta demanda clara de agilidad, el cumplimiento de una metodología del ciclo de vida de las aplicaciones, no es una barrera para conseguirlo. De hecho, más del 70% de las empresas admite seguir una metodología de este tipo específica, y el 82% indica que los procesos de pruebas se cumplen de manera consistente en el 50% o más de sus proyectos. Ello demuestra que la tendencia de “agile testing” y los procesos de pruebas no son excluyentes.

3.2.9 Los grupos de pruebas

Los grupos de pruebas se especializan en los procesos de verificación y validación (V&V), dando autonomía al proceso de pruebas dentro de un proyecto y compartiendo una estructura paralela al grupo de desarrollo [HOLOWAY99]. Los procesos de pruebas requieren técnicos independientes al desarrollo del software, que comprendan los errores y puedan proponer soluciones [IEEE98]. Esta independencia es importante para dar con errores sutiles que no son apreciables desde el lado del desarrollador [BOEHM84].

Los grupos de pruebas independientes

La estructura organizativa de los grupos de pruebas independientes, se basa en la existencia de un grupo de pruebas paralelo al grupo de desarrollo, que reporta con independencia a la gerencia. De esta forma, el gerente de pruebas no informa al gerente de desarrollo, ya que poseen el mismo peso dentro de la organización [ARTHUR00].

Un grupo de pruebas independientes es un grupo cuyo principal objetivo son las pruebas. Pueden estar asignados a las pruebas de un solo producto o de varios. De acuerdo con esto, cada grupo va a tener un jefe o un líder, cuyo interés y atención se

centra en el trabajo de su equipo. Así, se podrá dirigir un proceso de pruebas diseñado por técnicos ajenos al desarrollo, y además no se obtendrán presiones por parte del grupo de desarrollo ó presiones económicas. Obteniendo el grupo de pruebas el mismo peso que el resto de grupos que contribuyen en el producto.

Los estándares IEEE para la V&V software establecen la necesidad de utilizar grupos de pruebas independientes en desarrollos de sistemas críticos, formados por aquellos que pueden ocasionar pérdidas de vidas humanas o grandes pérdidas ya sean sociales o económicas [IEEE98].

Estos profesionales se centran en las pruebas, convirtiéndose con el tiempo en expertos en éste campo. Gracias a estos grupos, se llegaron a establecer las pruebas como una disciplina dentro de la ingeniería del software [CRAIG02].

Aunque la creación de estos grupos supuso una profesionalización y especialización de las pruebas, también supuso la creación de una barrera entre los grupos de probadores y desarrolladores. Los desarrolladores se vuelven menos inquietos al probar sus códigos desde que son conscientes de que los probadores van a realizar el trabajo de cualquier manera.

Actualmente uno de las mayores dificultades para los gerentes de los grupos independientes se trata de empezar su labor en fases tempranas del ciclo de vida de un producto, siendo esto un gran perjuicio, ya que la efectividad de las pruebas aumenta cuando antes se realicen en el ciclo de vida [CRAIG02].

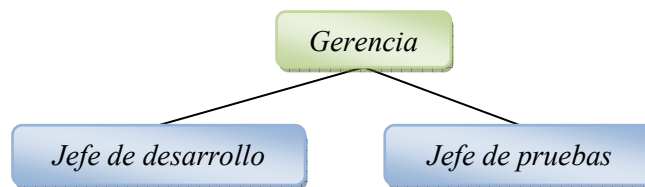


Figura 26. Estructura de los grupos de pruebas independientes.

Ventajas
La independencia de los probadores proporciona una especialización técnica.
La independencia directiva proporciona seguridad frente a presiones y ayuda centrarse en la calidad.
La garantía de objetividad.
Inconvenientes
La posible falta de comunicación entre equipos.
La posible pérdida de responsabilidad por la calidad.

Tabla 31. Ventajas e inconvenientes de los grupos de pruebas independientes.

Los grupos de pruebas independientes

La formación de grupos de pruebas dependientes implica la división de responsabilidades técnicas, pero no de gestión del proyecto [WALACE89]. Así, tanto la

parte de desarrollo como la de pruebas formarán una única estructura. Estructura utilizada normalmente por pequeños grupos de trabajo, de menos de 10 personas.

El grupo de pruebas informa al jefe de proyecto, que resulta ser la misma persona que gestiona el trabajo de los programadores [HOLOWAY99]. Esto puede ser un problema, al presentar tanto los programadores como los probadores los informes al mismo individuo, creando un inevitable conflicto de intereses. Por ello, es necesario un jefe de proyecto con experiencia, que su objetivo no solo sea solo desarrollar un software sino que además este sea de la máxima calidad. De este modo, tanto los programadores como los probadores deben ser considerados por igual. Esta estructura facilita los flujos de comunicación, al existir muy pocos niveles de gestión. Además, permite el trabajo mano a mano de programadores y probadores que puede llegar a ser muy eficiente [BOEHM84].

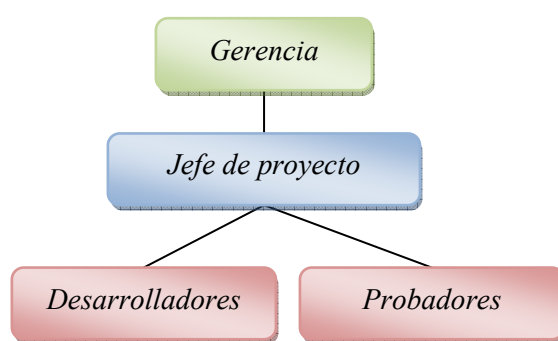


Figura 27. Estructura del grupo de pruebas dependiente.

Ventajas
La facilidad del flujo de comunicación. La especialización técnica. La familiarización con el proyecto.
Inconvenientes
El conflicto de intereses. La necesidad de un jefe de proyecto con gran experiencia.

Tabla 32. Ventajas e inconvenientes de los grupos de pruebas dependientes.

3.2.10 Perfiles profesionales de las TI

Un perfil profesional define la identidad de las personas que, con una titulación académica, llevan a cabo una determinada labor y, explica las funciones principales que dicha profesión cumple. La base para definir estos perfiles es el análisis de funciones y de tareas que llevan a cabo los profesionales teniendo en cuenta su posible futura evolución [MUT05].

Con estas premisas, e intentando abarcar las diferentes tendencias y propuestas profesionales existentes dentro del campo de las TI, en el “Computing Curricula 2001” de ACM e IEEE [ACM01] se diferencian el perfil profesional de desarrollo software, el

perfil profesional de sistemas y el perfil profesional de gestión y explotación de tecnologías de la información.

Perfil de desarrollo software

Un ingeniero en informática con un perfil profesional en desarrollo software debe estar capacitado para poder realizar tareas de análisis del producto, de estudio de soluciones, de desarrollo, de verificación y de mantenimiento del mismo. Estas capacidades se verán desarrolladas en competencias específicas, estableciendo actividades que no solo abarcan un punto de vista informático, sino que también el de la organización y la gestión software [DUTTA99], como son:

- Gestión de proyectos.
- Gestión del riesgo en software.
- Gestión de la calidad del software.
- Gestión de configuración.
- Gestión del proceso de desarrollo.

Por tanto, deben poseer un conocimiento amplio de las metodologías y herramientas de desarrollo, de SI (sistemas de información), SGBD (sistemas de gestión de bases de datos) y de herramientas para la automatización del propio desarrollo [SEI99].

En el desarrollo de este perfil profesional se pueden englobar otros más especializados, como son los propuestos por el consorcio Career Space [OVD] y por la Conferencia de la Profesión de Ingeniero e Ingeniero Técnico en Informática (COPITI) [COPITI]: Desarrollo de software y aplicaciones, arquitectura y diseño de software, producción e ingeniería del software o diseño multimedia.

Perfil profesional de sistemas

Un ingeniero en informática con perfil profesional de sistemas debe estar capacitado para analizar, diseñar, construir e implementar sistemas, que soporten aplicaciones técnicas, comerciales, industriales y de negocios en general. Utilizando técnicas y métodos que aseguren su eficiencia. Asimismo, administra los centros de cómputo o de sistemas de información de datos.

Por ello, debe contar con sólidos conocimientos de las técnicas, los dispositivos y las herramientas propias del ámbito, las cuales le capaciten para la especificación, el diseño, el montaje, la depuración, el mantenimiento y la evaluación del rendimiento del hardware de computadores y de sus periféricos habituales. También, debe ser competente en el desarrollo del software del sistema que posibilita una gestión eficaz de los recursos hardware del sistema informático.

Teniendo en cuenta la gran importancia que poseen hoy en día los sistemas distribuidos, el profesional de sistemas debe contar con grandes conocimientos en redes telemáticas de cualquier tecnología y/o extensión; en sistemas y procedimientos que proporcionen coordinación, seguridad y confidencialidad a todo el sistema; políticas de seguridad, tanto en la red como en los sistemas que interconecta; y en dispositivos físicos de red, medios y protocolos de transmisión y de los sistemas operativos.

En el desarrollo de este perfil profesional se pueden englobar otros más especializados, como son los propuestos por el consorcio Career Space [OVD] y por la Conferencia de la Profesión de Ingeniero e Ingeniero Técnico en Informática (COPITI) [COPITI]: Ingeniería de comunicación de datos, diseño de redes de comunicación, asistencia técnica o ingeniería de integración y pruebas e implantación y pruebas.

Perfil de gestión y explotación de TI

Un ingeniero en informática con perfil profesional de gestión y explotación de tecnologías de la información es responsable de asegurar que las necesidades de gestión de la información y del conocimiento de las organizaciones se satisfacen en el desarrollo y en la implantación de soluciones informáticas.

Por ello, debe conocer la estrategia empresarial y las diferentes soluciones tecnológicas necesarias para apoyar dicha estrategia. Lo que también supone conocer las últimas tendencias y tecnologías del sector de TIC.

Debe coordinar la labor de los otros ingenieros en informática con perfil profesional tanto de desarrollo software como de sistemas, para que esté alineada con los objetivos estratégicos de la organización.

En definitiva, es corresponsable de los resultados de la organización y será evaluado sobre esa base. Estos resultados se expresan en términos de satisfacción de los clientes, productos vendidos, servicios prestados y beneficios generados.

Este perfil ofrece la posibilidad de formar parte de los equipos directivos en el caso de estar integrados en una determinada organización o, en el caso de ser profesional independiente, será un consultor externo altamente especializado en las tecnologías de la información y de las comunicaciones.

Por lo general habrán ocupado antes puestos técnicos que posiblemente les hayan llevado a liderar equipos y proyectos, empezando allí a asumir responsabilidades directivas. Un ingeniero en informática con perfil profesional de gestión y explotación de tecnologías de la información puede llevar a una persona a los más altos niveles de una organización, como consejero delegado o director gerente.

Bajo este perfil profesional se abarcan los perfiles propuestos por el consorcio Career Space [CARRER] y COPITI [COPITI]: consultoría de empresas de TI, especialista en sistemas, dirección de marketing de TIC, dirección de proyectos de TIC, dirección de TIC, ingeniería de Sistemas y Consultoría de empresas de TI.

3.2.11 Competencias dentro de las TI

La definición de competencias en las TIC asume un papel importante a la hora de formar futuros trabajadores y establecer puestos de trabajo, caracterizándose porque comportan todo un conjunto de conocimientos, procedimientos, actitudes y rasgos que se complementan entre sí, de manera que el individuo debe “saber”, “saber hacer”, “saber estar” y “saber ser”, para actuar con eficacia frente a situaciones profesionales [ANECA]. Éstas solo pueden ser definidas en un determinado marco de trabajo, por lo que para su desarrollo adquieren especial importancia tanto la experiencia como el contexto en el que se demandan. Siendo consideradas como un concepto integrador, debido a que consisten en un conjunto de aptitudes y actitudes, de modo que van más allá de los componentes técnicos, los cuáles se complementan con los componentes metodológicos, participativos y personales. Por ello, un ingeniero no sólo debe saber lo que hay que hacer en una determinada situación, sino también debe ser capaz de extrapolar la información relevante con el fin de abordar cualquier situación. Asimismo, es un concepto dinámico porque las competencias se desarrollan a lo largo de la trayectoria profesional, es decir, no son inmunes a los cambios [ISUS01].

El proyecto Tuning [DEUSTO00], desarrollado dentro del marco del Espacio Europeo de Educación Superior (EEES) define las competencias desde la perspectiva de los resultados del aprendizaje como: “ los conocimientos, las habilidades, las actitudes y las responsabilidades, que describen los resultados del aprendizaje de un programa educativo o, lo que los alumnos son capaces de demostrar al final del proceso educativo”. Formando una de las bases del plan Bolonia.

Con el fin de estudiar las competencias en el ámbito de las TIC, se establecen dos tipos diferentes en función del ámbito al que aluden. Dividiéndose en competencias genéricas, las cuales engloban a la totalidad del sector, y en las competencias específicas, que forman las asociadas a un determinado perfil o rol profesional. Estas últimas, debido a su estrecha relación con los perfiles profesionales, van a ser desarrolladas más adelante ya que en los estándares previamente mencionados no se refleja la existencia de perfiles orientados a funciones de v&v, problemática a la que se intentará dar solución a lo largo del presente tema.

Las competencias genéricas, también conocidas como transversales, juegan un papel importantísimo en la formación del ingeniero en informática: constituyen el cemento que une todas sus competencias específicas, generando así una relación que establecerá la diferencia entre el saber y, el saber aplicar de manera óptima y con deontología profesional [MUT05]. Siguiendo el proyecto Tuning, se definen tres tipos de competencias genéricas: instrumentales, interpersonales y sistémicas.

Competencias instrumentales.

El objetivo principal de estas competencias es el de dotar a la persona con medios y métodos que le permitan utilizar sus conocimientos en el entorno en el que trabaja. Para un ingeniero, estas competencias son fundamentales ya que desarrollan su capacidad de organizar y de estructurar el trabajo, mucho más cuando de él se espera la dirección de los grupos a su cargo.

Se identifican las siguientes competencias instrumentales [TURLEY94]:

- **Reutilización del código:** El ingeniero intenta aprovechar proactivamente los esfuerzos realizados por otros, reutilizando códigos o sus diseños para crear un nuevo desarrollo de código.
- **Resolución metódica de problemas:** El ingeniero utiliza un enfoque metódico (construye modelos mentales, los diseña, experimenta, desarrolla herramientas de prueba, etc.) para la comprensión y solución de problemas.
- **Habilidad técnica:** El ingeniero es partidario uso de técnicas de diseño, competencias de depuración, realiza elecciones de tecnología con facilidad, posee buena base técnica y de desarrollo software.
- **Automatización de pruebas:** El ingeniero utiliza técnicas de prueba adicionales durante el desarrollo del código para lograr un alto grado de fiabilidad.
- **Experiencia:** El ingeniero debe de poseer experiencia con un proyecto similar.
- **Formación necesaria:** El ingeniero busca activamente la formación necesaria para completar la tarea asignada.
- **Utilización de nuevos métodos y las herramientas:** El ingeniero busca mejorar el rendimiento o los resultados a través de la utilización de nuevas herramientas y métodos.
- **Capacidad de planificación y estimación:** El ingeniero posee una gran preocupación por los horarios y sus estimaciones.
- **Utilización de prototipos:** El ingeniero utiliza la creación de prototipos para evaluar los parámetros fundamentales del sistema antes de diseñar el producto final.
- **Conocimientos relativos al ámbito de estudio:** El ingeniero debe poseer conocimientos relativos a las tareas asignadas.
- **Conocimientos de técnicas de comunicación estructuradas:** El ingeniero utiliza herramientas y técnicas de diseño estructurado con el fin de comunicar los diseños pero no sigue el formalismo completo de los enfoques.

Competencias interpersonales

El trabajo del ingeniero se realizará normalmente en grupos, por tanto la capacidad de comunicación y de trabajo en equipo del ingeniero será un factor determinante de su éxito y, de sus posibilidades de ascenso en una organización jerárquica. La comunicación de las soluciones a los equipos de trabajo (competencias sociales) con los que colabora el

ingeniero es una de sus labores principales, sus soluciones no sólo deben ser técnicamente correctas sino, además comprensibles.

También en este caso es difícil resaltar alguna competencia sobre las demás, pero está claro que el comportamiento ético (competencias individuales) será fundamental para su formación, no sólo como ingeniero, sino como persona, y las capacidades sociales serán imprescindibles para tratar con grupos de personas heterogéneos.

Se identifican las siguientes competencias personales [TURLEY94]:

- **Motivación por el deseo de contribuir:** El ingeniero es fomentado por los valores de la sensación de logro que viene de hacer una contribución directa.
- **Orgullo por la calidad y la productividad:** El ingeniero se enorgullece en producir productos libres de defectos en la fecha prevista en un tiempo mínimo.
- **Sentido de la diversión:** El ingeniero disfruta el desafío de la misión y el sentido de la realización de completarlo, se divierte en el trabajo.
- **Falta de ego:** Al ingeniero no le importa donde una buena idea viene y no siente la necesidad de promover sus propias ideas.
- **Perseverancia:** El ingeniero posee la disciplina, la tenacidad, la dedicación y voluntad para trabajar duro en una tarea.
- **Pro actividad:** El ingeniero toma la iniciativa para terminar tareas.
- **Amplitud de miras:** El ingeniero ve la situación en general en lugar de centrarse en los detalles.
- **Exhaustividad:** El ingeniero se asegura que todos los caminos están cubiertos es metódico, organizado, y cautelosos.
- **Responsabilidad:** El ingeniero es impulsado por su responsabilidad y expresa claramente los objetivos que a de lograr para conseguir un resultado específico.
- **Fuertes convicciones:** El ingeniero demuestra fuertes creencias y convicciones. Actúa de acuerdo con estas creencias, incluso cuando están en contra de la dirección específica de la gestión.
- **Mezcla objetivos personales y de trabajo:** El ingeniero persigue sus objetivos de proyecto con sus propios objetivos de desarrollo personal, y presiona a sus gerentes para recibir las asignaciones de trabajo que respondan a sus deseos personales.

- **Acepta ayuda:** El ingeniero debe saber pedir ayuda para el aprendizaje, la investigación, el diseño, la comprensión, la depuración, o para la comprensión los resultados de las revisiones.
- **Ayuda a otros:** El ingeniero debe emplear tiempo en ayudar a otros compañeros en la realización de sus tareas.
- **Superación:** El ingeniero no se satisface con acontecido en el momento, establece grandes expectativas y objetivos personales para progresar.
- **Trabajo en equipo:** El ingeniero comparte los valores del trabajo en equipo para crear soluciones de grupo a expensas de los resultados individuales.
- **Capacidad de discusión:** El ingeniero es capaz de enfrentarse a otra persona con el fin de resolver un problema.

Competencias sistémicas

Por último, se destacan las competencias sistémicas. Un ingeniero debe ser capaz de entender el conjunto en el que desarrolla su labor más allá de los simples objetivos inmediatos, por ejemplo el contexto social y económico del sector en el que trabaja.

En este sentido es necesario tener una actitud emprendedora que permita ver posibilidades y definir objetivos en entornos reales, además de disponer de capacidad de dirección y liderazgo para encaminar estos objetivos.

Se destacan las siguientes competencias de sistema [TURLEY94]:

- **Calidad:** El ingeniero preocupa por la fiabilidad y tiene un compromiso de alta calidad.
- **Visión de cliente:** El ingeniero debe considerar a los clientes o al usuario final y el intercambio de impresiones con ellos a de ser un ingrediente esencial en el diseño de productos.
- **Razonamiento:** El ingeniero debe poseer la capacidad de pensar algorítmicamente y estructuradamente.
- **Soluciones sencillas y elegantes:** El ingeniero crea soluciones que son elegantes y sencillas y permiten una fácil extensión para futuras necesidades.
- **Creatividad:** El ingeniero ha de tener ideas creativas.
- **Atención al detalle:** El ingeniero tiene la capacidad para hacer frente a la complejidad.

3.2.12 La verificación y la validación bajo el marco de las TI

Las organizaciones, especialmente las de mediano y gran tamaño, basan con cada vez mayor asiduidad sus negocios en sistemas de TI. Este hecho, tal y como se viene tratando a lo largo del proyecto, sumado a la creciente importancia de factores como: la complejidad, la heterogeneidad y la variabilidad de las tecnologías; ha supuesto que las pruebas del software y de los sistemas de información sean consideradas como una necesidad en la actualidad. Suponiendo además, un aumento en los presupuestos destinados a actividades de prueba, y en la demanda de servicios relacionados con las mismas en los últimos años [ELC07].

Dicha situación lleva a suponer que, debido a la cada vez mayor frecuencia con la que se implantan procesos y tecnologías informáticas a través de las diferentes normativas y estándares existentes, en busca de una mayor calidad del software, los puestos de trabajo orientados a desarrollar las labores necesarias para satisfacer dichas necesidades se encontraran definidos en profundidad, siendo además éstas actividades, donde se localizan las actividades de v&v que ocupan este estudio. Pero, como se ha podido observar en el marco que se ha presentado al inicio del tema a través de publicaciones como: *“los perfiles de capacidades profesionales genéricas de TIC de Career Space”* [CARRER], *“el libro blanco de ANECA”* [ANECA] y el proyecto Tuning [DEUSTO00], no solo no se realiza un especial hincapié en dichas labores, sino que ni siquiera se tratan aspectos de calidad genéricos. Lo cual conduce a que en el mercado laboral no exista una demanda de puestos orientados a la v&v con capacidades claramente definidas, tal y como se verá reflejado más adelante. Sólo en el caso de organizaciones muy especializadas en labores de calidad se produce una regulación de la actividad del sector, pero orientado al uso interno de la propia empresa.

En vista de la poca documentación obtenida al respecto y dado el amplio estudio que se viene realizando a lo largo del proyecto acerca de las labores de v&v, a continuación se ofrece una propuesta que pretende recoger las competencias y los perfiles profesionales que se enmarcarían bajo dicho tipo de actividades.

Para lo cual, en primer lugar, se debe tener en cuenta que el proceso de V&V se encuentra distribuido a lo largo del proceso de calidad de un desarrollo software. Por lo que, es lógico pensar que va a compartir una gran cantidad de aspectos en común. No obstante, como se acaba de mencionar, éste tampoco se encuentra desarrollado en profundidad. Por lo tanto, como los perfiles y las competencias específicas de otras áreas presentes en las TIC no aportan información relevante para el desarrollo de este tipo de competencias, serán descartadas, estableciendo como referencia para este tipo de perfiles las denominadas competencias genéricas, detalladas previamente a lo largo del punto 3.2.11. Debido a que éstas comprenden aptitudes comunes a las distintas áreas de las TIC, y como en este caso el proceso de V&V se encuentra orientado a dicho ámbito, deberá compartirlas de la misma forma.

Con el fin de desarrollar la propuesta que se viene mencionando a lo largo de este punto, una vez definidas las competencias genéricas que actuarían como base en el desarrollo de cualquier actividad de ingeniería, es necesario centralizar la propuesta. Por ello, cobra importancia seleccionar la información relativa a cómo se desarrolla el proceso en cuestión y así poder establecer las competencias que se van a encargar de ello. Pero antes, con el fin de establecer un marco que sirva de referencia a la hora de tratar los

perfiles profesionales orientados a la v&v, se hace referencia a los presentados por Tmap Next y MTP dado su alto grado de especialización en el proceso de prueba y a los diferentes perfiles que se solicitan en el mercado laboral en España, ya que pueden resultar, a posteriori, muy útiles a la hora de desarrollar la propuesta.

3.2.12.1 Marco profesional en TMap

Sogeti es una de las empresas con mayor peso en la calidad del software a nivel mundial. Tal es su especialización al respecto, que hace gala de una extensa metodología (TMAP), que hoy en día es tomada como referente. En ella, se establece una profunda segmentación de perfiles y de roles profesionales. Los cuales van a ser detallados a lo largo de este capítulo.

En Tmap, las actividades de pruebas dentro de un proyecto o dentro de una línea organizacional están definidas por distintas tareas, siendo necesarias unas habilidades en particular para llevarlas a cabo. Un grupo de tareas, junto al conocimiento y a las habilidades que requieren es conocido como un rol [DRIEL]. Éstos a su vez, forman parte de un concepto más amplio denominado posición, concepto que a lo largo del capítulo se ha denominado con el término de *perfil profesional*.

Con el fin de poder describir las distintas funciones y puestos orientados a la calidad que se pueden desarrollar en una organización, TMap propone una distinción entre las posiciones y los roles que se pueden ejercer. Siendo su principal diferencia que un profesional tiene una posición o perfil profesional que le enmarca dentro de un determinado tipo de actividades dentro de la organización y un determinado rol dentro del proceso, describiendo la posición una serie de características específicas que se deben de realizar dentro de dicho proceso.

Perfiles profesionales

Las posiciones o perfiles profesionales en TMap se establecen a través de un modelo propio de carrera profesional basado en tres niveles, por el cual se establecen:

Posiciones junior, en las cuales se supone cierta experiencia realizando pruebas:

- Probador.
- Programador de herramientas de prueba.

Posiciones intermedias, en las cuales se supone que quien desempeña la actividad es un experto en la realización de pruebas y que posee cierta experiencia en labore de asesoría:

- Experto en métodos de prueba.
- Coordinador de pruebas.
- Experto en herramientas de prueba.

Posiciones sénior, en las cuales se supone que quien desempeña la actividad es experto en funciones de prueba y asesoría:

- Consultor de pruebas.
- Gestor de pruebas.
- Consultor de herramientas de prueba.

Ésta clasificación permite establecer a la organización que implementa Tmap qué grado de experiencia y de madurez dentro de la empresa representa una determinada posición o perfil profesional. Proporcionando una guía para la evolución dentro de la organización, lo cual es un concepto fundamental a la hora de definir un perfil profesional.

Además, siguiendo con la línea que se viene desarrollando hasta ahora, Tmap plantea el establecimiento de las posiciones en función de un conjunto de competencias específicas que delimitan la actividad a desarrollar y que actúan como un factor diferenciador frente a los diferentes perfiles, estableciendo por cada uno:

- **Probador:** El probador ejecuta las tareas primarias de las pruebas. Las tareas de las pruebas son más creativas, pero también las más laboriosas e intensivas.

Competencias específicas:

- Las revisiones de comprobabilidad de las especificaciones de las pruebas.
 - La especificación de casos de prueba, las situaciones de salida o los *scripts* de prueba.
 - La creación situaciones de salida.
 - La ejecución casos de prueba.
 - La comprobación y el aseguramiento de los resultados de las pruebas.
 - La realizar de comprobaciones e investigaciones.
 - El registro de los defectos.
 - La conservación del programa de pruebas.
- **Programador de herramientas de prueba:** El programador de herramientas de pruebas es responsable de los aspectos técnicos del conjunto de las pruebas. El programador implementa el diseño dentro del marco de instrucciones dado por el consultor de herramientas de pruebas.

Competencias específicas:

- La realización, la gestión y del mantenimiento del conjunto pruebas.
 - El soporte en el diseño del conjunto de pruebas.
- **Experto en métodos de prueba:** El experto en métodos de pruebas es un especialista en las pruebas, ya sea de un determinado tipo de pruebas o de un

entorno. Posee experiencia y cumple con un conjunto de roles de ejecución y de asesoramiento dentro del proyecto, aunque también puede actuar como un líder de equipo.

Competencias específicas:

- La realización de prueba.
 - El soporte a la creación de planes de prueba.
 - El soporte a la creación de la organización de las pruebas.
 - El reclutamiento y la evaluación de miembros del proyecto.
 - La distribución del trabajo y el seguimiento del proceso.
 - El desarrollo y el mantenimiento de la formación de las pruebas.
 - La formación del grupo de trabajo.
 - La asesoría sobre la implementación de técnicas de pruebas.
 - La asesoría sobre al coordinador de pruebas.
 - La recopilación y la consulta datos estadísticos de las pruebas.
- **Coordinador de pruebas:** El coordinador de pruebas lidera el equipo de probadores, expertos en métodos de pruebas, programadores de herramientas de prueba y expertos en herramientas de pruebas. Siendo el responsable de la planificación, gestión y ejecución del proceso de pruebas. Realiza informes sobre el avance del proceso sobre el plan de pruebas y de la calidad de los objetos probados. Su función es constante durante todo el ciclo de vida de las pruebas.

Competencias específicas:

- La creación, la obtención, la aprobación y el mantenimiento del plan de pruebas.
 - La ejecución del plan acorde a su planificación y presupuesto.
 - La dirección de las consultas internas.
 - La asistencia a reuniones de proyecto.
 - El aseguramiento el proceso de pruebas.
- **Experto en herramientas de pruebas :** Un experto en herramientas de pruebas se trata de un programador que puede utilizar herramientas especiales o trabajar en determinados entornos. Posee una gran experiencia y una completa combinación de roles de ejecución y aseguramiento en los proyectos. También puede realizar funciones de líder de equipo para los programadores. Además, puede ayudar al consultor en la introducción de nuevas herramientas.

Competencias específicas:

- La realización de tareas como programador.
- El soporte en la creación de un conjunto automatizado de pruebas.
- El soporte en análisis rápidos.
- La configuración herramientas de prueba.
- El reclutamiento y la evaluación de empleados
- La distribución del trabajo y el seguimiento del proceso.

- El desarrollo y el mantenimiento de la formación de herramientas.
- La enseñanza y la formación.
- La mediación con la formación externa.
- El asesoramiento al coordinador de pruebas.
- **Consultor de pruebas:** El consultor de pruebas es experto en métodos de pruebas. Se centra exclusivamente en el trabajo de consultaría en relación con los aspectos de las pruebas. Ofrece soporte a los métodos orientados al proceso de pruebas.

Competencias específicas:

- La configuración de técnicas de pruebas.
- El desarrollo de nuevas técnicas de pruebas.
- La regulación las pruebas.
- La preparación de las formaciones.
- La formación del equipo.
- La asesoría y el soporte a la implementación de todos los tipos de técnicas de pruebas.
- La mejorar el proceso de pruebas.
- La consultoría de gestión.
- **Gerente de pruebas:** El gerente de pruebas lidera un equipo formado por coordinadores. Es el responsable de la planificación, el mantenimiento y la ejecución del proceso de pruebas según la planificación, el presupuesto y la calidad acordada. Realiza informes conforme se produce el avance del proceso de pruebas a través del plan y de la calidad de los objetos de prueba.

Competencias específicas:

- Las tareas propias del coordinador de pruebas.
- La creación, la obtención de la aprobación y el mantenimiento del plan de pruebas principal.
- La ejecución del plan de pruebas principal dentro de los márgenes planificados y presupuestados.
- El mantenimiento de los contactos externos.
- La gestión de la calidad interna.
- El reconocimiento, la anticipación e informe de los riesgos del producto.
- El consejo y la definición de la política de pruebas.
- La ejecución de la optimización del proceso de pruebas.
- Establecimiento de conocimiento requerido y competencias.
- **Consultor de herramientas de prueba:** El consultor de herramientas de prueba es un especialista que comenzó como un experto en pruebas. Se centra exclusivamente en el trabajo de consultaría en relación con los aspectos de las herramientas de pruebas. Ofrece soporte a los procesos relacionados con el desarrollo de herramientas.

Competencias específicas:

- La creación y la configuración del conjunto de pruebas.
- El desarrollo del nuevo conjunto de pruebas.
- La creación de regulaciones y normativas.
- El desarrollo de la formación relacionada.
- La formación del grupo de trabajo.
- La asesoría y el soporte para la implementación de todo tipo de herramienta.
- La definición de una política de pruebas.
- La ejecución de análisis rápidos.
- La consultoría de la gestión.

Roles

Tmap otorga una especial importancia al concepto de rol, si bien hasta ahora se ha venido considerando como un concepto de apoyo para dar a un perfil profesional una mayor especialización dentro de una organización evitando su equiparación. En Tmap el concepto de rol se centra en las actividades de prueba que son realizadas dentro de un proyecto específico de la organización. Por ello, forman un conjunto de competencias mucho más especializadas que los recogidos en las posiciones o perfiles profesionales, pero así mismo, al ser en el ámbito de una misma organización se encuentran fuertemente relacionados. Llegando a existir, roles que se corresponden con las posiciones mencionadas en el apartado anterior, como son [DRIEL]:

- Tester.
- Programador de herramientas de pruebas.
- Experto en métodos de prueba.
- Coordinador de pruebas.
- Experto en herramientas de pruebas.
- Consultor de pruebas.
- Gerentes de pruebas.
- Consultor de herramientas de pruebas.

Por el contrario, existen otro tipo de roles que no se asemejan a ninguna posición por sí solos. Esto viene dado fundamentalmente por su mayor especialización. Además de los presentados a continuación, en la práctica es posible que surja la necesidad de utilizar otros distintos, por lo que es posible desarrollar otros nuevos acordes con las necesidades. En Tmap Next por ejemplo, propone una serie de roles que no concuerdan en su totalidad con ninguna posición a modo de guía, tal y como se pudo apreciar a continuación [DRIEL]:

- **Integrador de aplicaciones:** El integrador de aplicaciones es el responsable de la integración de las diferentes partes del sistema (programas, objetos, módulos y componentes) en un sistema operativo correctamente, y es capaz de situarse como un inspector de calidad a fin de supervisar los acuerdos alcanzados. El integrador de aplicaciones apoya a la unidad de pruebas y lleva a cabo la unidad de integración de pruebas. Además prepara el sistema para la siguiente fase del proyecto.

Competencias específicas:

- La integración de los distintos programas, objetos, módulos y componentes en el sistema.
 - La obtención de la aprobación y del mantenimiento de la unidad de integración del plan de pruebas.
 - La creación de los criterios de salida que una parte integrada del sistema debería tener para estar preparada para afrontar la siguiente fase.
 - El soporte a los programadores en la ejecución de las pruebas de unidad.
 - La dirección de la ejecución de la gestión de la configuración y la gestión de la versión.
 - La dirección de la ejecución de la gestión de defectos internos.
 - El contacto con cliente sobre los tipos de pruebas posteriores.
 - La información del progreso de los procesos de integración y de la calidad de los objetos de prueba.
 - El asesoramiento sobre la unidad de pruebas de integración.
 - La evaluación de los procesos de integración.
 - La preparación del sistema para la siguiente fase.
- **Administrador de defectos:** El administrador de defectos es responsable de configurar y optimizar el uso de la administración de defectos y de los entornos de los procesos.

Competencias específicas:

- La configuración de la administración de los defectos y de los procesos asociados.
 - El asesoramiento sobre la selección de herramientas para la administración de defectos.
 - El seguimiento, la evaluación y la mejorar de los procesos y de los métodos en administración de defectos.
 - La gestión de los registros de defectos mediante el análisis y los foros de toma de decisión.
 - La administración del acceso a las herramientas de la administración de defectos.
- **Experto en el conocimiento del dominio:** El experto en el conocimiento del dominio ofrece un conocimiento sobre la materia y una visión de cómo el dominio está especificado en los requerimientos, diseños funcionales, procesos de negocio y en los manuales de usuario.

Competencias específicas:

- El soporte y el asesoramiento sobre los procesos funcionales y de negocio.
- **Intermediario:** El intermediario mantiene el contacto a nivel operacional con el equipo de pruebas, con los desarrolladores y con los expertos en la materia. Por otro lado, explican los defectos y proporcionan datos complementarios si son necesarios. También suministran soluciones a los defectos de los equipos de pruebas. Este rol toma especial importancia en el caso de existir equipos de pruebas distribuidos o si existe algún tipo de relación formal con otra parte interesada, como en el caso del outsourcing.

Competencias específicas:

- La exposición de los defectos a los consultores de calidad y a los participantes en el foro de análisis.
- La filtración defectos duplicados.
- La suministración datos complementarios a los defectos.
- La dirección de la evaluación de los defectos cuando estos se pueden producir bajo otros entornos.
- La comprobación de las soluciones para el ajuste de la política y la integridad del proyecto.
- La exposición de la solución a los defectos.
- El seguimiento del estado de los defectos.
- El informe del progreso de los defectos.
- La asesoría a los órganos de consultaría sobre los defectos.
- **Experto de sistemas:** El experto en sistemas tiene la responsabilidad de explicar a los probadores la funcionalidad y la arquitectura técnica en general del sistema bajo el que prueban. Debe dar soporte a las pruebas de sistema, a las pruebas de aceptación funcional, a las pruebas de aceptación de usuario y a las pruebas de integración de sistema.

Competencias específicas:

- La exposición de la funcionalidad del sistema, también en relación con el proceso de negocio.
- La exposición de la arquitectura técnica del sistema.
- **Coordinador de infraestructuras de pruebas:** El coordinador de infraestructuras de pruebas es responsable dentro del equipo de pruebas del contacto con las partes encargadas de configurar y mantener un alto nivel de disponibilidad de la infraestructura de las pruebas (entorno de pruebas, herramientas de pruebas y el lugar de trabajo del probador). Este rol se desarrolla preferiblemente por una única persona, preferiblemente a tiempo parcial. Si no es posible encontrar la persona idónea para este rol las tareas pueden ser distribuidas.

Competencias específicas:

- El aseguramiento de la disponibilidad oportuna de los entornos de prueba.

- El aseguramiento de la disponibilidad oportuna de las herramientas de pruebas con la planificación y el control requeridos tan pronto como lo requieran las herramientas de ejecución de pruebas.
 - El aseguramiento de la disponibilidad oportuna de los puestos de trabajo.
 - La coordinación con otras partes de la infraestructura de las pruebas.
 - El asesoramiento sobre la resolución de defectos o incidentes en el área de la infraestructura de las pruebas.
 - El asesoramiento sobre el impacto de los cambios propuestos en la infraestructura de las pruebas.
 - El asesoramiento sobre las especificaciones de la infraestructura de las pruebas.
 - El asesoramiento sobre los costes del uso de una infraestructura de pruebas.
- **Administrador del proyecto de pruebas:** El administrador del proyecto de pruebas es responsable de la administración de la información y de la disponibilidad del proyecto de pruebas, incluyendo la información del progreso y la documentación del proyecto.

Competencias específicas:

- La recogida, la comprobación y el registro de la información del progreso.
 - La suministración de la información del progreso.
 - La recogida y la administración la documentación del proyecto.
- **Líder del equipo de pruebas:** El líder del equipo de pruebas es el responsable del día a día del trabajo del equipo de pruebas. Este rol se introduce si el equipo de pruebas se encuentra distribuido. Por lo cual, cada equipo es dirigido por un líder, aunque si el equipo de pruebas es muy pequeño este rol se combina con el rol de un probador.

Competencias específicas:

- La creación detallada de la planificación del equipo de pruebas.
 - El seguimiento e el informe del progreso del proceso de pruebas.
 - El informe de la calidad del objeto de pruebas.
 - La gestión de la resolución de problemas de infraestructura.
 - El informe de cuellos de botella y de sus soluciones.
 - La gestión diaria del equipo de pruebas.
 - La formación de probadores en el equipo de pruebas.
- **Administrador de los programas de pruebas:** El administrador de los programas de pruebas es el responsable del seguimiento y del cumplimiento de las regulaciones existentes en el área de los programas de prueba.

Competencias específicas:

- La configuración de los programas, los procedimientos y las regulaciones de las pruebas.
- La gestión del programa de pruebas.

- La configuración de la gestión del programa, de la configuración de los básicos, de los objetivos y de la infraestructura de las pruebas.
- Comprobación del cumplimiento con la regulación de programa de pruebas.

3.2.12.2 Orientación profesional de las pruebas en MTP

Métodos y Tecnología (MTP) es la consultora española líder en servicios de ingeniería y de calidad software, la cual llevará a cabo para Telefónica el mayor proyecto de calidad de software de Europa [ELC07]. Debido a su alto grado de especialización en la materia, al igual que con Tmap, se pretender tomar como referencia el marco profesional que desarrollan con el fin elaborar una propuesta de competencias y perfiles profesionales que cubran actividades de v&v.

En este caso, MTP no facilita toda la información relativa a dicha estructura. Por lo que de la información obtenida, se puede observar la ausencia del concepto de perfil profesional y el uso exclusivo de roles. La concepción de un rol en éste caso, es la misma que a lo largo del presente estudio se ha dado a un perfil profesional, por lo que de aquí en adelante se asumirán los roles desarrollados en MTP como perfiles profesionales en vista de analizarlos en la posterior propuesta

Cada uno de estos roles va a poseer ciertas actividades asignadas dentro de un proyecto, y para desempeñar dichas tareas van a ser necesarios una serie de conocimientos y de experiencia.

Una vez definidos los roles existentes, en MTP se establece la identificación de las tareas que puede llegar a realizar cada empleado (competencias específicas), siéndoles asignados posteriormente los roles que mejor puedan desempeñar. Por otra parte, es necesario señalar que dos roles pueden necesitar capacidades comunes, con lo que puede ser fácil pasar de uno a otro. Sin embargo, lo normal será que dos roles realicen tareas completamente distintas. Por lo tanto, para alcanzar la capacitación necesaria para un determinado rol, habrá que impartir unas acciones formativas, al tiempo que se dejan de lado otras, que capacitan para un rol distinto [ELC07].

Roles

Según MTP, es posible implantar una propuesta de roles mediante los cuales desarrollar el modelo de capacitación. Mediante una breve descripción de sus responsabilidades, esta organización presenta los siguientes roles [ELC07]:

- **Ejecutor de pruebas.** El ejecutor de pruebas ejecuta casos o escenarios de prueba y documenta la ejecución.
- **Ingeniero de pruebas.** El ingeniero diseña casos y escenarios de prueba, los documenta y los construye.

- **Jefe de proyecto de pruebas.** El jefe de proyecto planifica las pruebas, coordina el proyecto y resuelve los problemas técnicos que puedan surgir, encargándose también del cierre.
- **Consultor de pruebas.** El consultor puede realizar labores de consultoría en diversas áreas, como metodología, automatización, infraestructura o soporte de negocio.
- **Especialista en automatización de pruebas.** El especialista tiene un conocimiento alto de los aspectos “teóricos” de las pruebas para las que se usan sus herramientas. Tiene conocimientos de desarrollo y/o arquitectura y está capacitado para crear y gestionar un test harness. Maneja una o varias herramientas de pruebas.
- **Gestor de Calidad.** El gestor se encarga del control de la calidad de los entregables y de la definición y obtención de métricas de proceso y de producto.

3.2.12.3 Demanda actual de perfiles de calidad

Con el fin de completar la escasa información existente en la actualidad referente a las competencias profesionales en el sector de la calidad del software, y más en concreto en lo referente al proceso de v&v, se ha decidido indagar sobre la oferta actual de empleo orientado a dicha actividad. Estas búsquedas han sido realizadas a través de los portales más importantes de demanda y oferta de empleo en TI, como son: Infojobs, Trabajos.com y Tecnoempleo. Gracias a los cuales, se ha podido obtener una extensa lista de ofertas de trabajo.

Dado que realmente para este estudio no resultan tan importantes las características específicas que se detalla en cada oferta de trabajo, como lo es la obtención de un marco que englobe los perfiles que resultan más comunes, se ha decidido a continuación presentar una extrapolación de los resultados con el fin de representar las características comunes que se pueden dar en dicho marco:

- **Técnicos:** En este grupo de ofertas de trabajo destaca el interés de las empresas por perfiles con poca formación en calidad, pero con conocimientos técnicos del objeto de pruebas. Así mismo, las empresas parecen asumir la formación en materia de calidad.
 - Experiencia mínima: No Requerida.
 - Nivel alto de inglés.
 - Interés por las nuevas tecnologías.
 - Una base técnica es recomendada pero no imprescindible.
 - Conocimientos de sistemas operativos.
 - Conocimientos de bases de datos.
 - Conocimientos en lenguajes de programación.
- **Analistas:** Para este colectivo, las empresas buscan una experiencia demostrada en labores de calidad y el conocimiento de metodologías y técnicas, otorgando un menor peso al conocimiento técnico de labores de

desarrollo y, destacando como valor añadido el conocimiento de los procesos de negocio implicados en la actividad del proyecto.

- Más de 5 años en el sector Test Software / Calidad Software
 - Alto nivel de inglés
 - Conocimientos de la arquitectura del proyecto.
 - Experiencia en programación en los entornos utilizados.
 - Conocimiento de herramientas de pruebas de rendimiento y estrés.
 - Pro-actividad en la resolución de problemas.
 - Relaciones interpersonales y trabajo en equipo.
 - Comunicación escrita y oral.
- **Gerentes:** El grupo que aúna perfiles de gerencia es el que posee una mayor carga directiva, se destaca por la búsqueda de habilidades sociales y de comunicación. Sin embargo, en la mayoría de las ofertas centradas en este perfil no se distingue el requerimiento de ninguna formación especializada, solicitando:
 - Estudios mínimos: diplomaturas.
 - Experiencia mínima en el sector de al menos 3 años.
 - Dominio de idiomas.
 - Experiencia en la coordinación de equipos.
 - Experiencia en el sector de las telecomunicaciones.
 - Pro-actividad en la resolución de problemas.
 - Persona metódica y organizada.
 - Importantes dotes de relaciones interpersonales y trabajo en equipo.
 - Gran capacidad de comunicación escrita y oral.

Los resultados obtenidos en el sondeo presentan, en general, una gran deficiencia dada la amplitud de la materia a tratar y a la escasa especialización que reflejan. Haciendo evidente la falta de estandarización y requiriendo únicamente competencias genéricas como factores determinantes en un perfil, cuando a lo largo del capítulo se ha venido observando que estas simplemente son una base del perfil no un factor diferenciador.

3.2.12.4 Propuesta de un marco profesional

Una vez obtenida la suficiente información acerca de: las competencias genéricas en la ingeniería, los modelos aplicados en organizaciones especializadas, la demanda actual y además, del desarrollo de la propia actividad que se viene realizando a lo largo de este proyecto; es posible presentar un marco que englobe las competencias y perfiles profesionales orientados a la v&v con ciertas garantías.

En primer lugar, se establece la estructura que va a representar la propuesta y en la que se puede observar una clara influencia de Tmap así como de la propuesta inicial del tema:

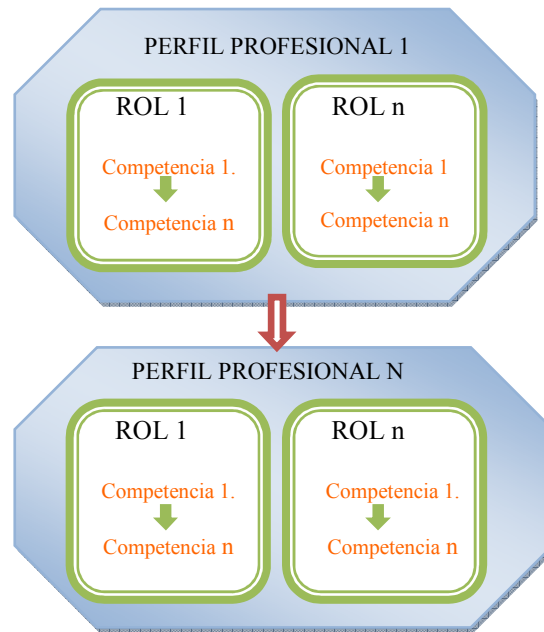


Figura 28. Propuesta de la estructura de un perfil profesional.

Con esta estructura se propone diferenciar los distintos conceptos que se manejan dentro del ámbito de las competencias, ya que como se ha podido comprobar, dependiendo de la fuente de información los conceptos puede variar, por ello:

- El *perfil profesional* se establece como un compendio de actividades y de características en forma de competencias que desarrollará el personal al que le sea atribuido dicho perfil dentro de una organización.
- Las *competencias* establecen el conocimiento de los profesionales, sus habilidades y sus actitudes en la realización eficaz de las tareas propias de un puesto de trabajo en un contexto concreto.

Como la adaptación de un perfil profesional a una determinada actividad puede resultar particularmente complicada si ésta necesita de unas competencias demasiado específicas, dado que la cantidad que se maneja en un perfil puede llegar a ser muy elevada, se produce un desglose en forma de **roles**. Éstos asumirán el papel de un perfil profesional con una mayor especialización y centrado en un determinado proyecto dentro de la empresa, permitiendo así seleccionar el personal más adecuado para una función específica dentro del abanico de posibilidades que presenta un perfil profesional. Pero la composición de roles entraría a formar parte de la propia organización de forma que su definición se establezca lo máximo posible al tipo de proyecto que se desarrolla.

Mientras que los roles no tienen porqué presentar ninguna relación, los perfiles profesionales establecen un concepto de evolución. Por lo cual, se realiza un reparto de competencias a modo de formación, que con la suma de experiencia en un determinado perfil permitirá la evolución a otro tipo de mayor responsabilidad.

Para establecer una guía que sirva de base a las diferentes competencias y perfiles se establece una referencia con el modelo TMMi, profundizado así en la funcionalidad del proceso.

Por lo tanto distribuyendo dichas características en función de una estructura jerárquica fácilmente reconocible y de común aplicación en el mundo de las TIC en la actualidad, se propone una distribución basada en perfiles orientados fundamentalmente a labores técnicas que implican poca responsabilidades y que permite una evolución paulatina al perfil más elevado centrado en tareas de gestión y con gran responsabilidad:

Perfil de dirección de pruebas.
Programa de formación de las pruebas. Medición de la prueba. Optimización de los procesos de prueba. Estrategia y política de pruebas. Organización de las pruebas. Entorno de pruebas.
Perfil de dirección de calidad de pruebas.
Control y supervisión de pruebas. Evaluación de la calidad del producto. Control de la calidad. Prevención de los defectos.
Perfil de analista de pruebas.
Diseños y ejecución de pruebas. Planificación de pruebas. Integración y ciclo de vida de las pruebas.
Perfil de probador.
Pruebas no funcionales. Revisiones por pares. Revisiones por pares avanzadas.

Tabla 33. Objetivos del proceso de pruebas por perfil profesional.

Estas tareas forman la base desde la que se va a desarrollar un determinado perfil, por lo que van a ser consideradas como parte de las competencias específicas de cada perfil profesional. Aunque por supuesto, cada uno de estos perfiles incorporará las competencias genéricas de la ingeniería detalladas al comienzo del tema.

Además como se ha podido observar en capítulos anteriores, la aplicación de un proceso de v&v se encuentra estrechamente relacionado con la aplicación de ciertas técnicas muy específicas y que serán fundamentales a la hora de distribuir competencias. Por lo tanto, se puede definir cada perfil:

El **perfil del probador** corresponde a un tipo de personal muy centrado en labores técnicas, por ello deberá poseer un conocimiento muy especializado en las distintas herramientas de pruebas y experiencia en su ejecución. Una serie roles que podrían surgir dentro de este tipo de perfiles son las propias especializaciones en técnicas de v&v. Así por ejemplo se establecerían roles como el probador de inspecciones, especialista en dicho tipo de técnica.

Además este tipo de perfil, según adquiere experiencia obtiene formación en la planificación y el análisis de pruebas dado su directa y constante implicación, lo que posibilita un progresión dentro del marco profesional.

El **perfil del analista** que se correspondería con el escalafón inmediatamente superior al de probador supone, como ya se venía indicando en éste último, un aumento en las responsabilidades y de las competencias en labores de análisis y planificación aunque sin dejar de lado la parcela más técnica. De tal forma que supone un paso intermedio entre las labores técnicas y las de dirección. Por ello, la existencia de distintos roles en este perfil no varía como en el caso de los probadores en función de la tecnología que manejen ya que para realizar sus funciones deben abstraerse de estos niveles, diferenciándose en función del objetivo al que se encuentren destinados. Así podrían surgir roles como el del analista del proceso de pruebas o el analista de técnicas de validación.

Todo esto viene a dar cada vez una mayor responsabilidad en labores de planificación y dirección lo cual implicaría un nuevo nivel.

El **perfil del director de calidad de pruebas** es el primer perfil donde establecería una diferenciación en función del objetivo al que se encuentre destinado el perfil, hasta ahora se venía realizando a través de roles, pero a estas alturas se debe tener un fuerte conocimiento del proceso al que se encuentra orientada la planificación. En este tipo de perfil en particular, se velará por la calidad del propio proceso y a partir de éste se definirán una serie de roles hasta llegar a los probadores. Estableciendo la planificación y dirección de dicho proceso, para lo cual se pueden distinguir distintos roles en función del número de recursos y proyectos a su cargo, por ejemplo un jefe de equipo que dirige un pequeño grupo en un solo proyecto, y un director de calidad del proceso que puede dirigir distintos jefes de equipo.

El **perfil de director de pruebas**, por el contrario, se encargará de la planificación y dirección del proceso de pruebas orientado a satisfacer la calidad del proyecto de desarrollo. La funcionalidad del perfil es muy parecida a la del director de calidad de pruebas incluso en sus distintos roles pero los objetivos que marcan su desarrollo es totalmente distinto.

Para completar esta propuesta se decidió asignar distintas competencias específicas con el fin de reflejar la definición de cada perfil que acaba de ser presentado. Es necesario destacar que propósito de este marco no es otro que el de resultar de base para el desarrollo de los distintos perfiles, por lo que, como se mencionó al comienzo es importante que cada organización pueda adaptarlo a su entorno estableciendo para ello la configuración de los distintos roles.

- **Competencias específicas del perfil del probador.**
 - Las revisiones de la gestión del software con el fin de evaluar el estado del trabajo realizado y de tomar decisiones con respecto a las siguientes actividades.
 - Las revisiones de la intervención *del software*.
 - La revisión del código que consiste en una forma de evaluación sistemática del código.
 - El desarrollo de pruebas por componentes.

- El desarrollo de pruebas no funcionales.
 - El desarrollo de revisiones por pares.
 - El desarrollo de revisiones por pares avanzadas.
 - Las revisiones de comprobabilidad de las especificaciones de las pruebas.
 - La especificación de casos de prueba, las situaciones de salida o los *scripts* de prueba.
 - La creación situaciones de salida.
 - La ejecución casos de prueba.
 - La comprobación y el aseguramiento de los resultados de las pruebas.
 - La realizar de comprobaciones e investigaciones.
 - El registro de los defectos.
 - La conservación del programa de pruebas.
 - La realización, la gestión y del mantenimiento del conjunto pruebas.
 - El soporte en el diseño del conjunto de pruebas.
 - Las revisiones de par del software, con el fin de evaluar el contenido técnico y/o la calidad del trabajo.
- **Competencias específicas del perfil del analista de pruebas.**
 - El desarrollo de revisiones formales.
 - La preparación de *inspecciones* para evaluar el código centrándose principalmente en los documentos, los modelos y el código.
 - La comprobación de los *walkthroughs*.
 - La revisión técnica de las pruebas.
 - Los diseños y la ejecución de las pruebas.
 - La planificación de las pruebas.
 - La integración del ciclo de vida de las pruebas.
 - El soporte a la creación de planes de prueba.
 - El soporte a la creación de la organización de las pruebas.
 - El desarrollo y el mantenimiento de la formación de las pruebas.
 - El desarrollo de nuevas técnicas de pruebas.
 - La interpretación de las especificaciones funcionales para la elaboración de casos de prueba.
 - La formación del grupo de trabajo.
 - La distribución del trabajo y el seguimiento del proceso.
 - La asesoría sobre la implementación de técnicas de pruebas.
 - La asesoría sobre al coordinador de pruebas.
 - La recopilación y la consulta datos estadísticos de las pruebas.
 - El reclutamiento y la evaluación de miembros del proyecto.
 - La ejecución del plan acorde a su planificación y presupuesto.
 - La dirección de las consultas internas.
 - La creación, la obtención, la aprobación y el mantenimiento del plan de pruebas.
 - La configuración de técnicas de pruebas.
 - La revisión de las actividades de ingeniería del software.
 - La auditoría de los productos software seleccionados.
 - La mejora del proceso de pruebas.
 - La consultoría de gestión.
 - La supervisión todas las etapas de la vida de un proyecto.

- El análisis y la recogida de nuevas técnicas y de herramientas del mercado, estudiando su viabilidad y necesidad.
 - El control y la gestión proceso.
 - El estudio de nuevas técnicas, métodos y herramientas de calidad.
 - La creación, la obtención de la aprobación y el mantenimiento del plan de pruebas principal.
 - La ejecución del plan de pruebas principal dentro de los márgenes planificados y presupuestados.
 - La gestión de la calidad interna.
 - El reconocimiento, la anticipación e informe de los riesgos del producto.
 - El consejo y la definición de la política de pruebas.
 - La ejecución de la optimización del proceso de pruebas.
 - Establecimiento de conocimiento requerido y competencias.
- **Competencias específicas del perfil del director de calidad.**
 - El control y la supervisión de las pruebas.
 - El planteamiento de una estrategia y de la política de pruebas.
 - El desarrollo de una organización del proceso de pruebas.
 - La evaluación de la calidad de las pruebas realizadas.
 - El control de la calidad del proceso de pruebas.
 - La prevención de posibles defectos en el proceso.
 - La preparación y el desarrollo de un entorno de pruebas óptimo.
 - El desarrollo de auditorías de las pruebas llevadas a cabo.
 - El asesoramiento a los usuarios, programadores y jefe de estudios en la redacción de la documentación de usuario, instalación y explotación.
 - El estudio de métodos, técnicas y herramientas de análisis y de diseño.
 - El estudio de la evolución de las nuevas tecnologías.
 - El diseño de la política de calidad.
 - La realización de estudios funcionales y de proyectos específicos.
 - La organización y distribución del trabajo de los equipos de v&v.
 - La dirección, la planificación y la coordinación de la gestión de la infraestructura de redes y comunicaciones.
 - La definición de los planes de formación y de reciclaje profesional.
 - El análisis de los proyectos y las necesidades y la sugerencia de soluciones en el plano técnico, humano y financiero.
 - La redacción de las propuestas comerciales que pueden implicar soluciones estándar o a medida.
 - La negociación de los contratos.
 - El aseguramiento de la definición de las directrices de calidad, su aplicación así como la estandarización.
 - Las pruebas de entregas con el fin de validar que el sistema satisface sus requerimientos y de asegurar que es fiable.
 - Las pruebas de aceptación para establecer si la entrega es lo suficientemente buena.
 - El desarrollo de las pruebas de rendimiento.

- **Competencias específicas del consultor de pruebas**
 - El desarrollo del programa de formación de las pruebas.
 - La medición de los resultados obtenidos en las pruebas.
 - La optimización de los procesos de prueba.
 - El desarrollo de las pruebas de rendimiento.
 - Las tareas propias del coordinador de pruebas.
 - La creación, la obtención de la aprobación y el mantenimiento del plan de pruebas principal.
 - La ejecución del plan de pruebas principal dentro de los márgenes planificados y presupuestados.
 - El mantenimiento de los contactos externos.
 - La gestión de la calidad interna.
 - El reconocimiento, la anticipación e informe de los riesgos del producto.
 - El consejo y la definición de la política de pruebas.
 - La ejecución de la optimización del proceso de pruebas.
 - Establecimiento de conocimiento requerido y competencias.
 - La configuración de técnicas de pruebas.
 - El desarrollo de nuevas técnicas de pruebas.
 - La regulación las pruebas.
 - La preparación de las formaciones.
 - La formación del equipo.
 - La asesoría y el soporte a la implementación de todos los tipos de técnicas de pruebas.
 - La mejorar el proceso de pruebas.
 - La consultoría de gestión.

3.3 Fase de obtención de resultados

Después de obtener las características más importantes que reflejan la adaptación a los procesos de verificación y validación software por parte de la información presentada a lo largo de la documentación, con el fin de establecer una guía útil para su implantación en una organización es necesario establecer que conceptos resultan más importantes para ello.

Una vez estudiada la necesidad de implantar un proceso de pruebas en una organización, o mejor dicho, una vez estudiado el nivel de madurez del proceso de pruebas que posee una organización es necesario tomar conciencia del objetivo de la propia organización respecto a este antes de afrontar cualquier tipo de medida.

Debido a que a medida que se ha ido desgranando el proceso de verificación y validación, se ha podido observar la amplitud del mismo. El primer factor que se deberá tener en cuenta se trata del tamaño de la organización en el que se quiere desarrollar dicho proceso.

Este factor puede llegar a ser especialmente limitante, ya que desde el primer momento en el que se considera un desarrollo de buenas prácticas, en este caso guiadas por CMMi, el potencial de la organización para llevar a cabo las tareas que se le supone puede ocasionar importantes restricciones y adaptaciones.

Aún así, se hace evidente la necesidad de implantar un modelo que cumpla con las características de CMMi o, por lo menos, tener en consideración la estrecha relación existente entre el proceso de verificación y validación y el resto de los procesos de una organización dedicada al desarrollo software. Lo cual promueve la adaptación de modelos por etapas que faciliten un desarrollo equilibrado.

Una vez que la organización adquiera ciertos niveles de madurez, puede permitirse la opción de fomentar ciertos procesos dentro de su desarrollo, los cuales le otorguen cierto valor de mercado. En este caso, como la atención se encuentra dirigida a las pruebas, cobran especial importancia los modelos de mejora de las mismas. Su aplicación guiará el desarrollo del proceso hasta niveles de madurez mucho más elevados que los que permite CMMi.

Por otra parte, se debe tener en cuenta que si el modelo de referencia sobre el que desarrolla el conjunto de los procesos es CMMi, y la intención es mantener y mejorar los niveles de madurez a través de su desarrollo, a la hora de implantar un modelo de mejora orientado al proceso de pruebas resulta cuanto menos interesante seguir la estructura planteada por este, tal y como posibilita TMMi.

Sin embargo, aunque a priori esta parece la mejor opción no hay que descuidar otros puntos de vista. La aplicación de TMMi marca pautas para el desarrollo del proceso de pruebas, las cuales además pueden ser llevadas a cabo bajo una estructura basada en etapas. Pero éste no desarrolla de manera exhaustiva el proceso de pruebas. De esta forma la metodología TMap puede ser un buen complemento, pudiendo ser tomada como referencia a la hora de implementar las pautas marcadas por TMMi. Ya que describe en profundidad la aplicación de un proceso de pruebas.

Simultáneamente, para poder llevar a cabo la implantación de dichos modelos, es necesaria la formación de equipos orientados a labores de calidad, tanto a nivel colectivo como a nivel individual. Debido a que la organización estructural delimita el tipo de profesional que va a ser solicitado y formado, ésta debe ser considerada con anterioridad. Este hecho que claramente plasmado en el nivel 3 de TMMi, en el área de proceso de “organizaciones de las pruebas”. En las cuales se definen las siguientes pautas que van a ser tomadas en cuenta para su desarrollo:

- **SG1.** Establecer una organización de pruebas.
 - **SP1.1.** Definir la organización de las pruebas.
 - **SP1.2.** Obtener compromisos de la organización de las pruebas.
 - **SP1.3.** Implementar la organización de las pruebas.
- **SG2.** Establecer funciones de las pruebas para los especialistas.
 - **SP2.1.** Identificar las funciones de las pruebas.
 - **SP2.2.** Desarrollar descripciones de los trabajos.
 - **SP2.3.** Asignar personal a las funciones de las pruebas.

- **SG3.** Establecer trayectorias de carreras de las pruebas.
 - **SP3.1.** Establecer trayectorias de las carreras de las pruebas.
 - **SP3.2.** Desarrollar planes personales de carreras de pruebas.

A la hora de definir las organizaciones de prueba, hoy en día parece premiar en el mundo de los negocios el factor de la reducción de costes. Siguiendo este factor a rajatabla, parece exagerado contar con recursos propios que velen por la calidad del producto y su desarrollo, pudiendo externalizar estas actividades en busca de un menor coste y de esta forma poder centrarse en el objetivo principal del negocio, el desarrollo. Por ello, en primer lugar la externalización de todo el proceso parece la mejor opción, pero por el contrario se ven marginados factores como el compromiso con la empresa, la facilidad de comunicación y la seguridad de ejercer un control y un seguimiento propio, tal y como queda plasmado desde los primeros niveles de TMMi, que a la larga pueden llegar a ser muy determinantes.

Llegados a este punto es recomendable la combinación de ambos, por una parte la subcontratación de recursos e incluso de actividades que doten al servicio de una mayor especialidad y economicidad, y por la otra la creación de recursos propios de control y supervisión, sin desdeñar la importación que para llevar a cabo un proceso de pruebas de calidad tiene que la independencia tanto directiva como técnica del mismo. Un ejemplo de ello sería la formación de grupos internos de SQA y la subcontratación de grupos independientes.

Pero existe un problema a la hora de implantar una organización de este tipo enmarcada bajo el desarrollo de TMMi. Ya que este modelo no contempla la externalización de recursos ni la gestión de los mismos.

Una vez definida la organización y asumidas las ventajas y desventajas propias de su implantación, es necesario establecer las funciones de las pruebas para el personal. El personal interno va necesitar de una serie de competencias de alto nivel orientadas a la gestión y al control del proceso, lo cual siguiendo la propuesta planteada en la documentación se centra en:

- **Perfil de dirección de pruebas.**
 - Programa de formación de las pruebas.
 - Medición de la prueba.
 - Optimización de los procesos de prueba.
 - Estrategia y política de pruebas.
 - Organización de las pruebas.
 - Entorno de pruebas.
- **Perfil de dirección de calidad de pruebas.**
 - Control y supervisión de pruebas.
 - Evaluación de la calidad del producto.
 - Control de la calidad.
 - Prevención de los defectos.

Este hecho conduce a que los proveedores de outsourcing se encarguen de la contratación de personal de alta cualificación técnica. Esto da a la empresa grandes

niveles de calidad técnica y un control y una administración de confianza, pero vuelve a chocar con el planteamiento de de TMMi. Ya que se define el establecimiento de carreras profesionales, las cuales de esta forma se ven drásticamente reducidas. Para poder implantar estas pautas en una empresa, a continuación se presenta una metodología para la implantación de un proceso de pruebas en una organización software basada en catorce fases:

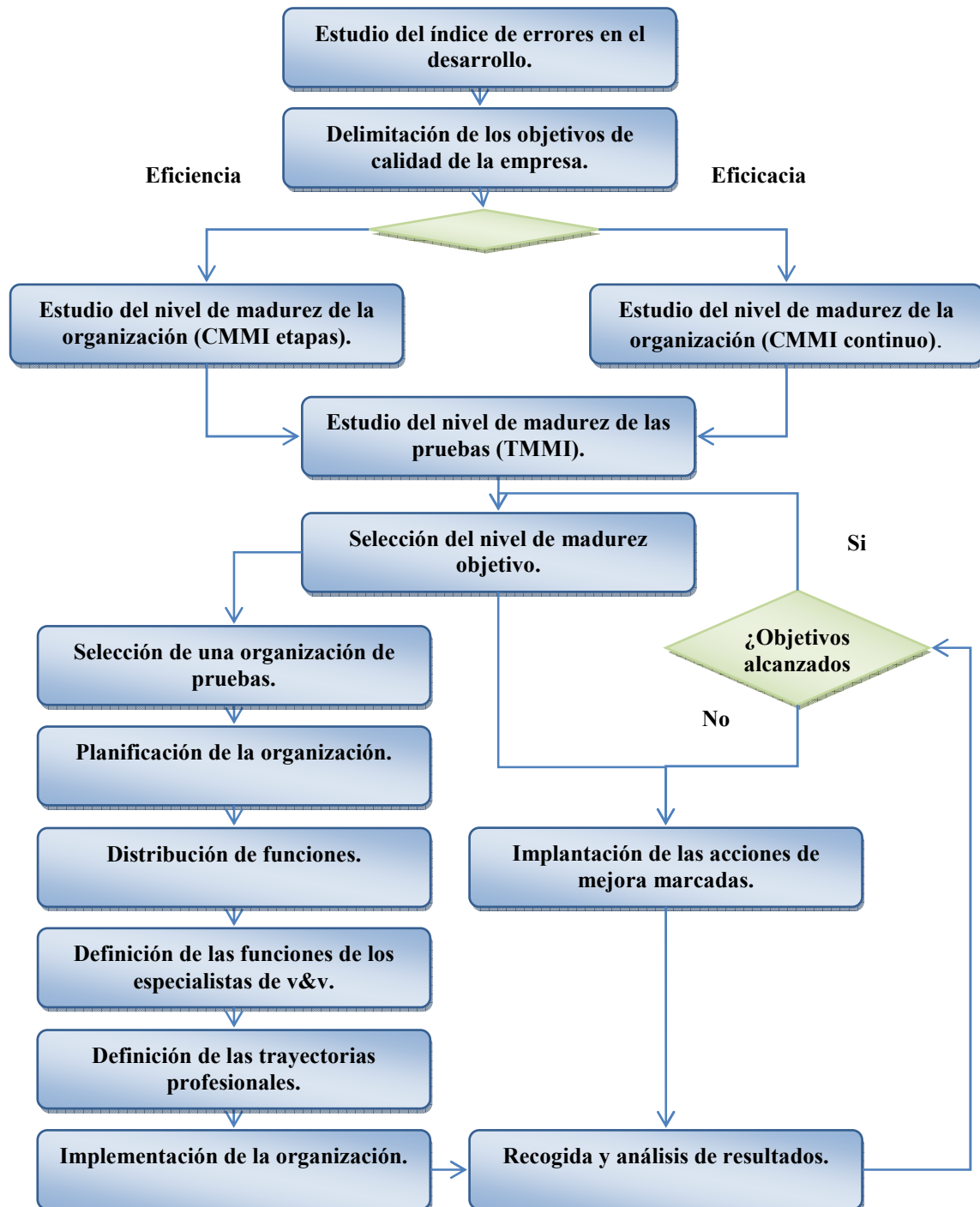


Figura 29. Metodología para la implantación de un proceso de v&v.

El estudio del indicie de los errores en el desarrollo es la actividad que se realiza en primer lugar. A través de la observación de los anteriores proyectos de una empresa es posible establecer dónde y en qué cantidad se suelen producir las desviaciones de los proyectos en función de su planificación.



Figura 30. Tareas del estudio del indicie de los errores en el desarrollo.

La delimitación de los objetivos de calidad de empresa marca el camino a seguir por la misma, estableciendo las metas que deben de ser alcanzadas durante el desarrollo del producto. De tal forma, que permite estudiar si estos se cumplen o no y que quedan expresados en función de CMMi.

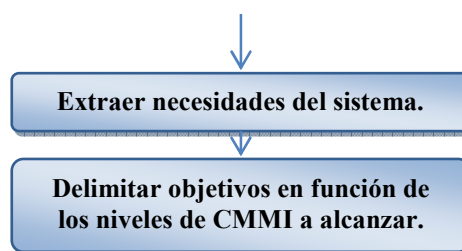


Figura 31. Tareas de la delimitación de los objetivos de calidad de empresa.

El estudio del nivel de madurez de la organización (etapas) es la fase mediante la cual una vez asumido que no se alcanzan los niveles de calidad y se pretende actuar al respecto desde el punto de vista de la eficiencia, mejorando así el conjunto de procesos de la organización.

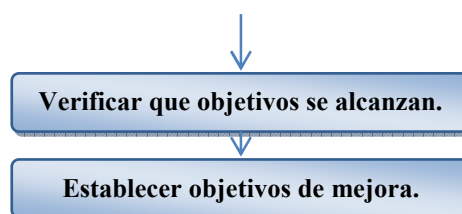


Figura 32. Tareas del estudio del nivel de madurez de la organización (etapas).

El estudio el nivel de madurez de la organización (continuo) es la fase mediante la cual una vez asumido que no se alcanzan los niveles de calidad y se pretende actuar al respecto desde el punto de vista de la eficacia, centrando así los esfuerzos de mejora en un solo proceso, en este caso la V&V.

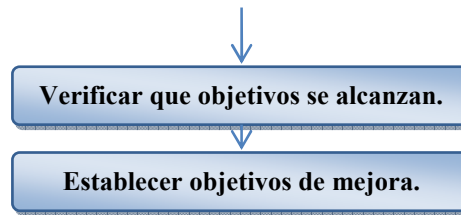


Figura 33. Tareas del estudio el nivel de madurez de la organización (continuo).

El estudio del nivel de madurez del proceso de pruebas es una fase clave en el proceso, ya que la v&v es el eje principal de la calidad y de esta forma, a través de su estudio mediante TMMi es posible establecer el nivel de desarrollo con el que cuenta el proceso de forma previa a realizar la mejora.

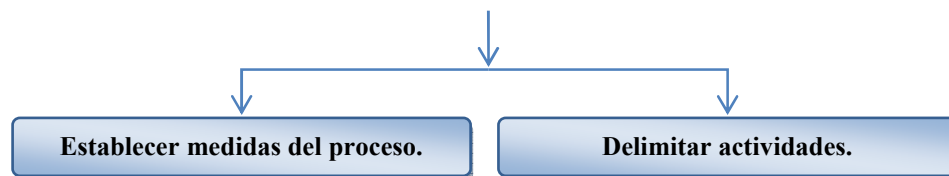


Figura 34. Tareas del estudio del nivel de madurez del proceso de pruebas.

La selección del nivel de madurez objetivo es la fase en la cual, una vez obtenido el nivel en el que se encuentra el proceso con anterioridad, se fijan los objetivos que desea alcanzar la empresa mediante la elección de niveles superiores a los actuales en TMMi.

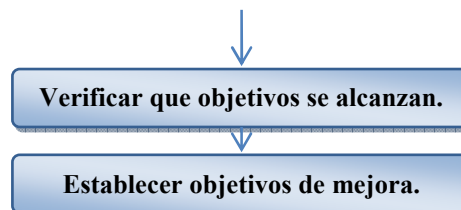


Figura 35. Tareas de la selección del nivel de madurez objetivo.

La implantación de las acciones de mejora es la fase en la cual se proceder a desarrollar las mejoras necesarias para alcanzar el nivel de madurez de TMMi objetivo.

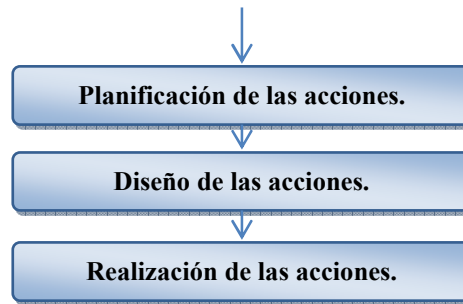


Figura 36. Tareas de implantación de las acciones de mejora.

La selección de una organización de pruebas es una fase paralela a la mejora del proceso, ya que sirve de soporte para su desarrollo y en la cual se estudian las mejores opciones organizacionales estructurales dentro de un proceso de v&v.

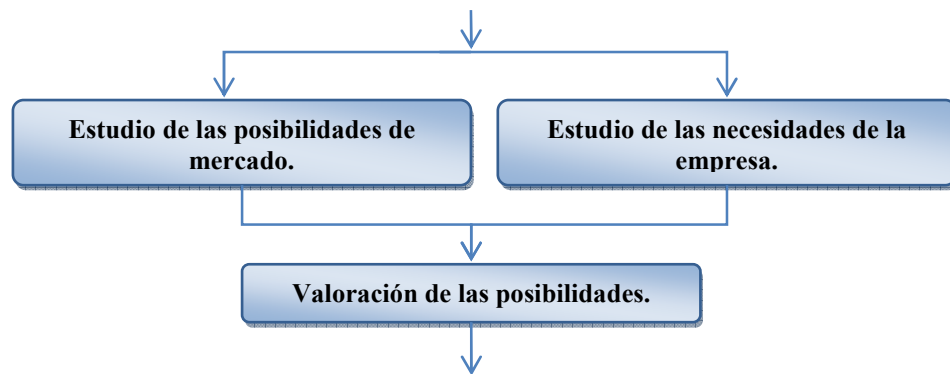


Figura 37. Tareas de la selección de una organización de pruebas.

La planificación de la organización de las pruebas es la fase que se lleva cabo una vez barajadas las distintas opciones que existen en el mercado a la hora de implementar una organización estructural, y en la cual se adaptan las organizaciones elegidas a las necesidades planteadas por la propia empresa.

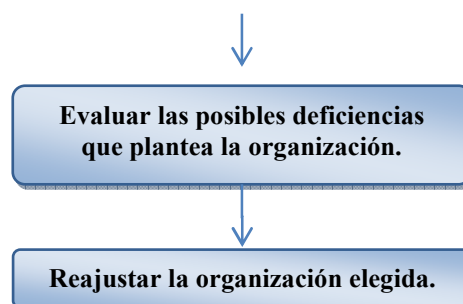


Figura 38. Tareas de la planificación de una organización de pruebas.

La distribución de funciones de la organización de las pruebas es la fase mediante la cual se procede a delimitar el marco de las actividades de la organización que se va a implantar, así como las responsabilidades que van a ser adquiridas por cada grupo que forme dicha organización.

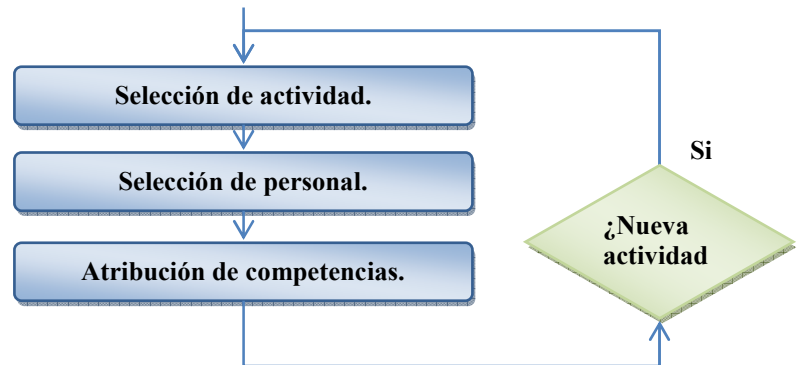


Figura 39. Tareas de la distribución de funciones de la organización de las pruebas.

La definición de las funciones de los especialistas de v&v es la fase mediante la cual se definen las diferentes competencias que van a ser atribuidas a los especialistas que van a contribuir en el desarrollo de proceso de v&v y su mejora.

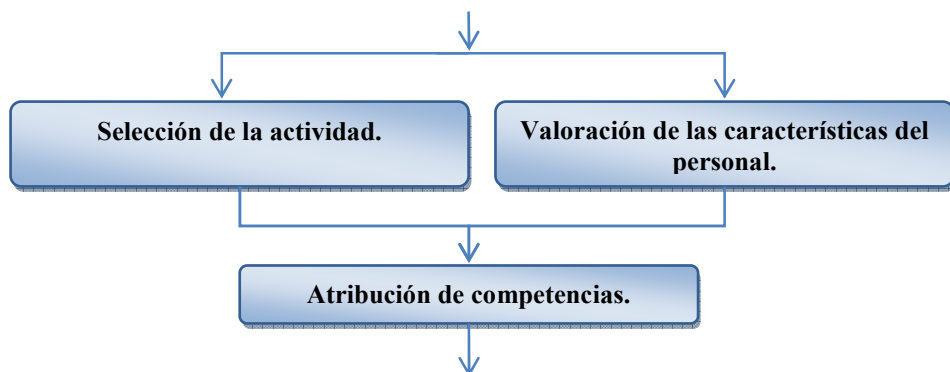


Figura 40. Tareas de la selección de una organización de pruebas.

La definición de las trayectorias profesionales es la fase donde se diseñan los diferentes perfiles profesionales, y con ello la posibilidad de que un empleado dedicado a la v&v pueda avanzar y tener un futuro dentro de la organización.

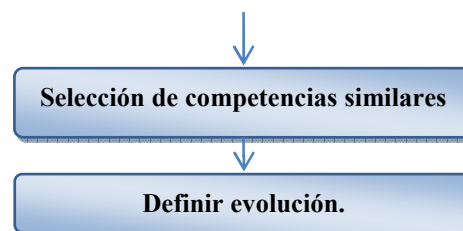


Figura 41. Tareas de la definición de las trayectorias profesionales.

Implementación de la organización de las pruebas es la fase donde se contratan y se pone en marcha la organización previamente definida.

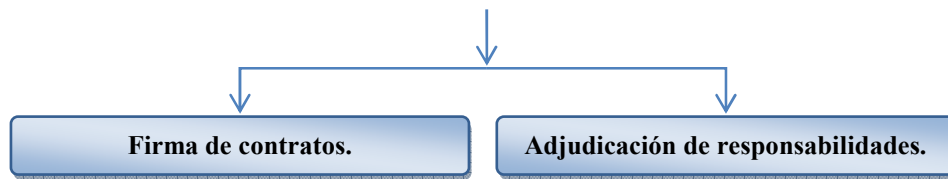


Figura 42. Tareas de la implementación de la organización de las pruebas.

Recogida y análisis de resultados es la fase final, donde se valoran los resultados obtenidos tras la implementación de la mejora del proceso de v&v, y donde se estudia la conveniencia de establecer nuevos niveles de madurez o seguir trabajando en la mejora del mismo.

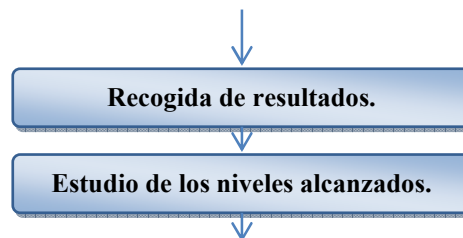


Figura 43. Tareas de la recogida y análisis de los resultados.

Capítulo 4

Conclusión y futuras líneas

4.1 Conclusión

Tras la consecución de los objetivos planteados en este proyecto, se ha obtenido gran cantidad de información acerca del estado de la calidad del software en el mercado de las Tecnologías de la información y de los procesos de prueba.

Gracias a la información disponible en este proyecto, una organización puede adecuar el desarrollo de su proceso de v&v en función de las necesidades de su organización. Para ello, dispone de resúmenes de los modelos más destacados del mercado TI, distintas estructuras organizativas y una propuesta de competencias que permite seleccionar y formar a su propio personal.

Como se ha podido observar, el proceso de pruebas ha sufrido un gran desarrollo desde sus inicios, cuando era considerado como una actividad marginal. La evolución que ha sufrido hasta nuestros días la establece ya como un proceso, con un ciclo de vida propio y sobre la cual se realizan importantes fases como la de diseño y planificación.

Además del desarrollo de la propia actividad, se ha producido una creciente aceptación por la comunidad, ya que resulta una labor imprescindible para obtener altos niveles de calidad, tanto del producto como de los procesos.

Pero si algo ha dejado claro la realización de este proyecto es, que a pesar de esta creciente aceptación, el arte de las pruebas aún se encuentra lejos de obtener altos niveles

de unidad y de adaptación. Un claro ejemplo de ello, es que no son pocos los modelos de mejora del proceso de pruebas y organizaciones diferentes existentes, sobre los se localizan grandes diferencias.

Este hecho, por una parte ofrece a las empresas una innumerable cantidad de formas de adaptar su proceso de pruebas de la manera que mejor le convenga, pero a su vez, supone que el desarrollo del proceso en sí sea mucho más lento y tedioso, ya que el avance del que pueda aportar un modelo no siempre es fácilmente exportable al resto.

Igualmente la gran cantidad de organizaciones existentes abre multitud de vías a las empresas para implementarlas, pero también limita la capacidad de formar al personal que se emplea en ellas. Debido a que estos normalmente cambian de proyecto, de empresas con la consecuente necesidad de afrontar cambios sustanciales en su formación lo que limita su aprendizaje y formación, respecto a que los puestos ofrecidos fuesen similares y afrontase los nuevos retos con una base mucho más fuerte.

De hecho, un claro ejemplo de ello es el de los modelos referentes a las competencias profesionales que se han presentado, ya que tan sólo en TMAP y en MTP se presentan competencias relacionadas. La ausencia total de estos factores tanto en la demanda actual como en los informes presentados por Career Space de muestra la falta de especialización en términos de pruebas y de calidad. Pero ni tan siquiera estas empresas llegan a cumplir con garantías con un perfil profesional claro y exportable. En TMAP se produce un exceso de especialización, es decir, a modo de implantar su metodología en una gran empresa, a priori ésta resulta óptima, pero a modo de referencia para el resto de organizaciones no ofrece la libertad necesaria para configurar su propio equipo. Implica demasiados perfiles para abarcar el conjunto de las competencias. Por otra parte, MTP sí refleja el proceso de pruebas entorno a unos perfiles más genéricos dentro del propio campo de las pruebas, lo que permite una adaptación al grueso de organizaciones para que éstas puedan reajustar los distintos perfiles a través de la asignación de roles específicos.

4.2 Futuras líneas

El desarrollo del proyecto ha buscado tratar las áreas que pueden resultar más relevantes para una empresa a la hora de implantar un proceso de verificación y validación. Pero tal y como ha quedado de sobra demostrado, la envergadura de este proceso hace que el número de factores a tener en cuenta para su implantación y desarrollo sea muy superior.

De hecho, un concepto que no se trata en este proyecto pero que abre una vía fundamental para la adaptación de las pruebas a una organización se trata del estudio de la calidad de las pruebas. Como se ha venido viendo las pruebas se encuentran íntimamente relacionadas con la calidad, debido a que éstas resultan una herramienta fundamental para obtener altos niveles de calidad tanto el productos como en los procesos de desarrollo software. Por esto mismo, al ser las pruebas un proceso también podrán ser controladas y medidas con el fin de obtener unas pruebas eficaces y eficientes. Por ello, el

estudio de las mediciones del propio proceso es una vía que presenta un gran interés para futuras investigaciones.

Además sería interesante investigar la problemática que se ha mencionado en el punto anterior, la falta de estandarización o la falta de aceptación de los estándares por la comunidad. Para lo cual, la presentación de un modelo que unificase el arte de las pruebas presentando las características más relevantes de los distintos modelos, organizaciones, competencias y métricas con el fin de establecer una guía de referencia para futuras implementaciones, suponiendo así un punto común de desarrollo y evolución.

Capítulo 5

Presupuesto

5.1 Resumen y diagrama de Gantt

Tal y como se ha venido desarrollando a lo largo de la propuesta, la realización de este proyecto fin de carrera dada su orientación al análisis de información se ha centrado en tres fases fundamentales:

- **La búsqueda de información**, donde se ha procedido a recopilar el grueso de la información con la que se ha desarrollado el proyecto, a través de:
 - Búsquedas.
 - Clasificaciones.
- **El análisis de la información**, donde se ha extraído la información más relevante para el tema en cuestión creando una primera valoración de la misma y facilitando la obtención de futuras conclusiones. Para lo cual se han realizado tareas de:
 - Localización.
 - Esquematización.
- **La obtención de los resultados**, donde a través del trabajo previo se extraen las valoraciones oportunas y se elabora la documentación final.
 - Elaboración de la documentación.
 - Revisiones.

5.1 Resumen y diagrama de Gantt

Con el fin de presentar el seguimiento llevado durante el desarrollo de dichas actividades se incluye el correspondiente diagrama de Gantt y un desglose de costes de personal, costes del material y costes totales.

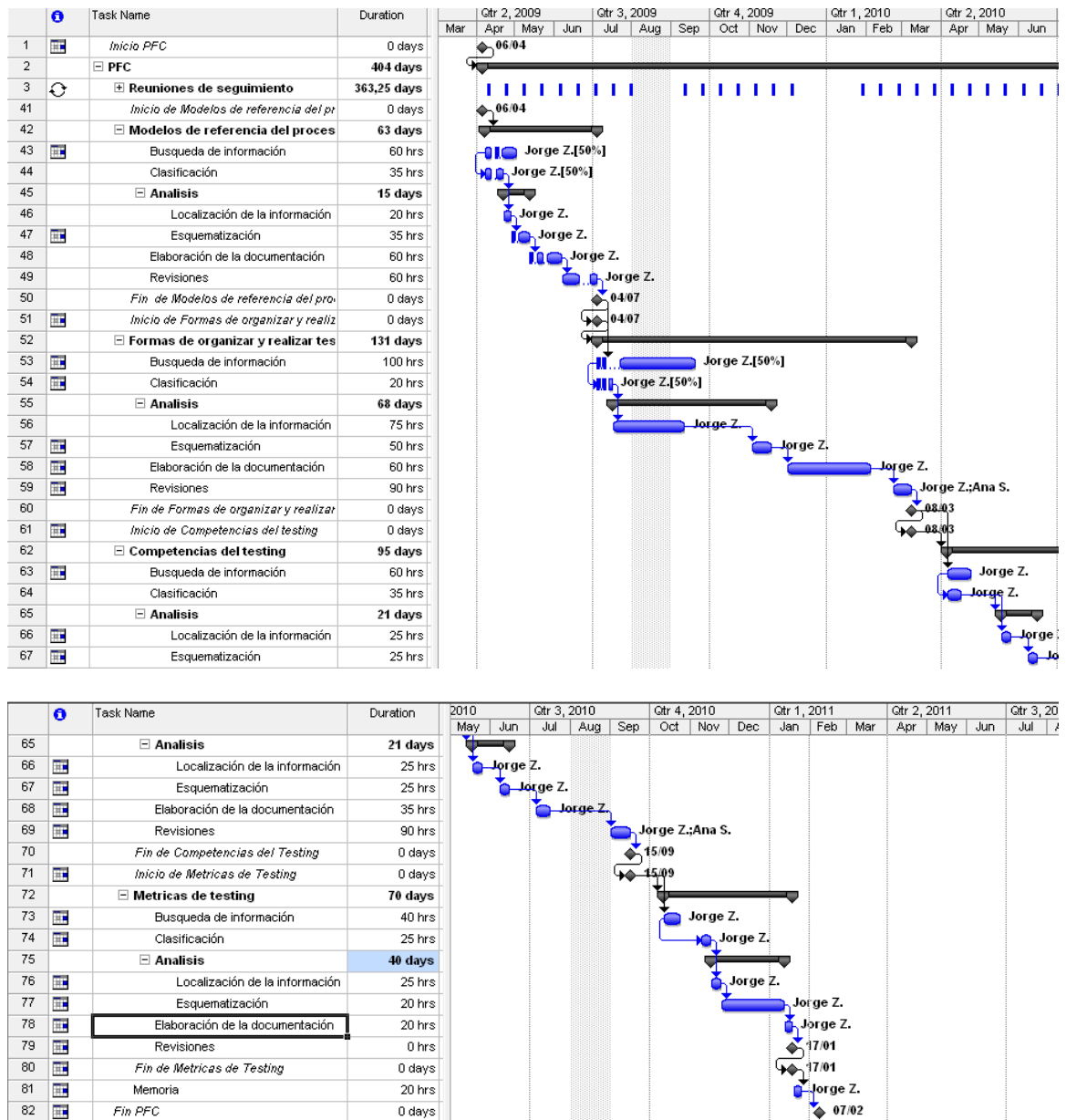


Figura 44. Diagrama de Gantt.

CAPÍTULO 5: Presupuesto



PRESUPUESTO DE PROYECTO

1.- Autor:

Jorge Zamora Hernandez

2.- Departamento:

Ingeniería del software.

- Titulo

- Duración (meses)

Tasa de costes indirectos:

Análisis de los procesos de verificación y validación en las organizaciones software

15

20%

4.- Presupuesto total del Proyecto (valores en Euros):

Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación mes) ^{a1}	(hombres	Coste hombre mes	Coste (Euro)	Firma de conformidad
Sanz Esteban, Ana		Ingeniero Senior		4	4.289,54	0,00	
Zamora Hernandez, Jorge		Ingeniero		8	2.694,39	17.158,16	
						21.555,12	
						0,00	
						0,00	
Hombres mes 12					Total	38.713,28	

^N 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^g
Ordenador portátil	700,00	70	18	60	147,00
Ordenador sobremesa.	1.000,00	30	12	60	60,00
		100		60	0,00
		100		60	0,00
		100		60	0,00
					0,00
Total					207,00

^{d)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%).

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
	Total	0,00

OTROS COSTES DIRECTOS DEL PROYECTO^{e)}

Descripción	Empresa	Costes imputable
	Total	0.00

^{a)} Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,....

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	38.713
Amortización	207
Subcontratación de tareas	0
Costes de funcionamiento	0
Costes Indirectos	7.784
Total	46.704

Figura 45. Presupuesto.

“El presupuesto total de este proyecto asciende a la cantidad de CUARENTA Y SEIS MIL SETECIENTOS CUATRO EUROS.

Leganés a 31 de Enero de 2011

El ingeniero proyectista

Fdo. Jorge Zamora Hernández.

Glosario

ANSI	<i>American Nacional Standards Instituye.</i>
AP	<i>Áreas de proceso</i>
BDTM	<i>Gestión de pruebas impulsadas por la empresa.</i>
CMM	<i>Modelo de Capacidad y Madurez.</i>
CMMI	<i>Modelo de Capacidad y Madurez Integrado.</i>
CEP	<i>Centros expertos en pruebas.</i>
DoD	<i>U.S. Department of Defense</i>
FP	<i>Factoría de prueba.</i>
GG	<i>Objetivo generico.</i>
IV&V	<i>Verificación y validación Independiente.</i>
IAOP	<i>Asociación Internacional de Profesionales de Outsourcing.</i>
NDIA	<i>National Defense Industrial Asociación.</i>
PDCA	<i>Circulo de Deming.</i>
PMLC	<i>Ciclo de vida del proyecto.</i>
SEI	<i>Software Engineering Institute.</i>
SDLC	<i>Ciclo de vida del desarrollo .</i>
SG	<i>Objetivo específico.</i>
SO	<i>Subobjetivos de madurez.</i>
SQA	<i>Garantía de calidad del software.</i>
TI	<i>Tecnología de la información.</i>
TMap	<i>Aproximación a la gestión de pruebas.</i>
TMM	<i>Modelo de Madurez de Pruebas.</i>
TMMI	<i>Modelo de Madurez de Pruebas Integrado.</i>
TPI	<i>Mejora del Proceso de Pruebas.</i>
V&V	<i>Verificación y validación.</i>

Referencias

[SCHIJNDEL] Ton van Schijndel, Dré Robben, “*TPI, CMMI and Testing*”, Sogeti Nederland B.V.

[VEENENDAAL06] Erik van Veenendaal, “*Guidelines for Testing Maturity: The Test Maturity Model*”, STEN Journal, Vol. IV, 2006

[GOSLIN08] Andrew Goslin, Klaus Olsen, Fran O’Hara, Mac Miller, Geoff Thompson, Brian Wells ,*Test Maturity Model Integration (TMMi)*,TMMI Foundation 2008, Editor: Erik van Veenendaal

[KOOMEN99] Sogeti Nederland B.V, “*Mejora del proceso de pruebas usando TPI*”, Koomen, T., Pol, M. “Test Process Improvement: a Practical Step-by-Step Guide to Structured Testing”, editorial Addison Wesley Longman, 1999, ISBN 0-201-59624-5

[AALST06] van der Aalst; Broekman, Bart y Vroonl,Michie,”*TMap ®Next for result-driving testing*”, UTN Publishers,Tim,Koomen,2006 ISBN:9072194802

[DRIEL] Ewout van Driel, “*Software Control y Testing TMap Next® Pruebas impulsadas por los objetivos de negocio*”, SOGETI España.

[BLACK] Rex Black, “*Advanced Software Testing Vol. 2: Guide to the ISTQB Advanced Certification as an Advanced Test Manager*”, Safari Online Books, Publisher: Rocky NookPub Date: December 28, 2008, Print ISBN: 978-1-933952-36-9, Web ISBN: 1-933952-36-9, pp 357-397.

[DUTTA99] Soumitra Dutta, Michael Lee, and Luk Van Wassenhove, “*Software Engineering in Europe: A Study of Best Practices*”, IEEE Software, vol. 1, no. 3, June 1999, ISSN:0740-7459 pp. 45 – 53.

[MYERS79] Glenford J. Myers, Tom Badgett, Todd M. Thomas, Corey Sandler, “*Art of Software Testing*”, John Wiley & Sons, Inc., New York, NY, 1979

[KOOMEN99] Koomen, T., Pol, M. (1999), “*Test Process Improvement: a Practical Step-by-Step Guide to Structured Testing*”, Addison-Wesley, ISBN 0-201-59624-5.

[KARINSALO] Mikko Karinsalo and Pekka Abrahamsson, “*Reuse: Methods, Techniques*”, ISBN978-3-540-22335-1, pp 59-68

[PYSTER] Arthur Pyster, “*What Beyond CMMI Is Needed to Help Assure Program and Project Success?*”, Springer, Berlin, Lecture notes in computer science ISSN 0302-9743

[SCHAEFFER04] Schaeffer, Mark: “*DoD Systems Engineering and CMMI. CMMI Technology Conference and User Group*”, <http://www.dtic.mil/ndia/2004cmmi/CMMIGS/SchaefferCMMI17Nov04v3.pdf>.

[KASSE] Kasse, T., “*Practical Insight into CMMI*”. 2004: Artech House, ISBN 1-58053-625-5

[SPR] “*A Framework for Adopting Software Process Simulation in CMMI Organizations*”, SpringerLink, ISSN0302-9743, DOI10.1007/978-3-540-72426-1, ISBN978-3-540-72425-4 , pp 320-331

[DUMKE] Ayaz Farooq and Reiner R. Dumke, “*Developing and Applying a Consolidated Evaluation Framework to Analyze Test Process Improvement Approaches*”, Institute for Distributed Systems, University of Magdeburg, ISSN0302-9743 ,ISBN978-3-540-85552-1, DOI10.1007/978-3-540-85553-8, pp 114-128

[VEENENDAAL] Erik van Veenendaal, “*Test Techniques in practice - Do they help? Why do we often test without them?*”, tmmiFoundation, ISSN 1866-5705

[SWINKELS] Ron Swinkels, “*A comparison of TMM and other Test Process Improvement Models*”, Frits Philips Institute

[SWAN96] Suwannasart, T. (1996), “*Towards the development of a Testing Maturity Model*”, Diss. Illinois Institute of Technology, March 1996

[KNEU08] Ralf Kneuper, “*CMMI: Capability Maturity Model Integration A Process Improvement Approach*”, Publisher: Rocky NookPub Date: December 28, 2008, Print ISBN: 978-1-933952-28-4, Web ISBN: 1-933952-28-8, Web ISBN: 1-933952-28-8, Safari Online Books.

[PERSSE06] James R. Persse, PhD, Publisher: O'Reilly Media, “*Process Improvement Essentials*”, Inc.Pub Date: September 14, 2006, Print ISBN-10: 0-596-10217-8, Print ISBN-13: 978-0-596-10217-3, Pages: 352, Safari Online Books.

[DENNIS03] Dennis M. Ahern; Aaron Clouse; Richard Turner, “*CMMI® Distilled: A Practical Introduction to Integrated Process Improvement*”, Second Edition, Publisher:

Addison-Wesley Professional Pub Date: September 23, 2003, Print ISBN-10: 0-321-18613-3, Print ISBN-13: 978-0-321-18613-3

[SHRUM06] Mary Beth Chrissis, Mike Konrad, Sandy Shrum, “*CMMI®: Guidelines for Process Integration and Product Improvement*”, Second Edition, Publisher: Addison-Wesley Professional Pub Date: November 13, 2006, Print ISBN-10: 0-321-27967-0, Print ISBN-13: 978-0-321-27967-5, Web ISBN-10: 0-321-63583-3, Web ISBN-13: 978-0-321-63583-9, Pages: 704, Safari Online Books

[SEI06] CMMI Product Team, “*CMMI for Development, Version 1.2, CMMI-DEV v1.2*”, CMU/SEI-2006-TR-008, Technical Report, Software Engineering Institute, August 2006, URL:<http://www.sei.cmu.edu/pub/documents/06.reports/pdf/06tr008.pdf>

[DAUGH02] Taz Daughtrey, “*Fundamental Concepts for the Software Quality Engineer*”, 288 pages, ASQ Press, ISBN 0-87389-521-5, 2002. <http://books.google.es/books?hl=es&lr=&id=dFmXBaLy0YMC&oi=fnd&pg=PA181&dq=tmm+test+maturity+model&ots=uXmmjSuOI&sig=Ecy2DNltHVZJnx3I0yUMC2pzBsQ#v=onepage&q=tmm%20test%20maturity%20model&f=false>

[ANDER04] Jari Andersin, “*TPI – a model for Test Process Improvement*”, Helsinki, 5th October 2004, Seminar on Quality Models for Software Engineering, Department of Computer Science, <http://www.cs.helsinki.fi/u/paakki/Andersin.pdf>, UNIVERSITY OF HELSINKI.

[FAROOQ08] Ayaz Farooq, Reiner R. Dumke, “*Evaluation Approaches in Software Testing*”, Nr.: FIN-05-2008, Arbeitsgruppe Softwaretechnik, http://www.cs.uni-magdeburg.de/fin_media/downloads/forschung/preprints/2008/TechReport5.pdf

[KALYANI99] Hossein Saiedian, Kalyani Chennupati, “*Towards an evaluative framework for software process improvement models*”, doi:10.1016/S0164-1212(99)00034-5, Copyright © 1999 Elsevier Science Inc. All rights reserved, Department of Computer Science, University of Nebraska at Omaha, Omaha, NE 68182-0500, USA, Metamor Information Technology Services, 4000 McEwan Road South, Suite 200, Dallas, TX 75244, USA

[MARTIN02] Pol Martin, “*Software testing : a guide to the TMap approach*”, Editor Addison-Wesley, 2002, ISBN 9780201745719.

[TMAP] “*Essentials of TMap Next*”, http://eng.tmap.net/Home/TMap/The_4_essentials/index.jsp

[SEI] Software Engineering Institute, <http://www.sei.cmu.edu/>

[FREIDKES07] Ruben Freidkes H, “*El Modelo TPI*”, Ubicación:<http://www.calidaddelsoftware.com/eventos/SoloPrueba2007/Ponencias/SOGETI%20-%20El%20Modelo%20TPI.pdf>

[BUMSTEIN03] Ilene burnstein, “*Practical software testing*”, Springer; 1 edition (June 24, 2003), ISBN-10: 0387951318, ISBN-13: 978-0387951317.

[BOEHM84] Boehm, B.W. 1984. “*Verifying and Validating Software Requirements and Design Specifications*”, IEEE Software, January: 75-88.

[IEEE98] IEEE 1998. “*IEEE Standards for Software Verification and Validation*”, IEEE Standard 1012-1998.

[HOLLOWAY99] Arthur, J.D., M.K. Groener, K.J. Hayhurst, and C.M. Holoway 1999. “*Evaluating the Effectiveness of Independent Verification and Validation*”, IEEE Computer 32(10): 79-83.

[WALACE89] Wallace, D.R. and R.U. Fujii 1989. “*Software Verification and Validation: An Overview*”, IEEE Computer 6(3): 10-17. ISSN: 0740-7459.

[ARTHUR00] James D. Arthur, Richard E. Nance , “*Varification and Validation Without Independence: A recipe for a failure*”, Pages: 859 – 865, Year of Publication: 2000, ISBN:0-7803-6582-8,Society for Computer Simulation International San Diego, CA, USA.

[AMESC02] Gonzalo Cuevas Agustín, VV Staff, Antonio de Amescua Seco, Centro de Estudios Ramón Areces, “*Gestión del proceso software*”. Editor Editorial Ramón Areces 2002, ISBN8480045469,9788480045469, Google books.

[BLACK03] Rex Black, “*Critical Testing Processes: Plan, Prepare, Perform, Perfect*”, Publisher: Addison-Wesley Professional, Pub Date: July 29, 2003, Print ISBN-10: 0-201-74868-1, Print ISBN-13: 978-0-201-74868-0. Safari Online Books.

[CORRIV07] Jean-Pierre Corriveau, “*Testable Requirements for Offshore Outsourcing*”, Editor: Springer Berlin / Heidelberg, ISSN: 0302-9743 (Print), 1611-3349 (Online), Volumen: Volume 4716/2007, LibroSoftware Engineering Approaches for Offshore and Outsourced Development , DOI10.1007/978-3-540-75542-5 , Copyright 2007, ISBN978-3-540-75541-, Páginas27-43

[TINN01]Paul C Tinnirello , “*New Directions in Project Management*”, ISBN: 978-0-8493-1190-1, Published by: Auerbach Publicatio, Publication Date: 26/09/2001, Google Books

[MELLON02] Anandasivam Gopal ,Tridas Mukhopadhyay, Carnegie Mellon, Mayuram S. ,Ann Arbor, “*The Role of Software Processes and Communication in Offshore Software Development*”,Pages: 193 – 200, Year of Publication: 2002, ISSN:0001-0782,ACM New York, NY, USA.

[BOEHM81] Boehm, B.W. “*Software Engineering Economics*”. Prentice-Hall Inc., Englewood Cliffs, NJ, 1981.

[FERG04] Ernest Ferguson, “*Impact of offshore outsourcing on CS/IS*”, Pages: 68 – 77, Year of Publication: 2004, ISSN:1937-4771,Publisher Consortium for Computing Sciences in Colleges , USA.

[CAPAC] "Executive Survey: The Outsourcing Institute's Annual Survey Of Outsourcing End Users," http://www.capacityllc.com/research/top_ten.pdf.

[DIBB04] Jens Dibbern, Tim Goles, Rudy Hirschheim, Bandula Jayatilaka, *Information Systems Outsourcing: A Survey and Analysis of the Literature*, Pages: 6 - 102 , Year of Publication: 2004, ISSN:0095-0033, ACM New York, NY, USA.

[GILL05] Nasib S. Gill , *Factors affecting effective software quality management revisited*, Pages: 1 – 4, Year of Publication: 2005, ISSN:0163-5948, ACM New York, NY, USA.

[ANDREWS04]James H. Andrews, *Relevant Empirical Testing Research: Challenges and Responses*, Pages: 1 - 4 ,Year of Publication: 2004, ISSN:0163-5948 ACM New York, NY, USA.

[BOGER84] Dan C. Boger, Norman R. Lyons, “*The organization of the software quality*”, Pages: 11 - 15 , Year of Publication: 1984, ISSN:0095-0033, ACM New York, NY, USA.

[CSAE] Consejo Superior de Informática, “*La calidad del equipo lógico*”, <http://www.csae.map.es/csi/silice/Auditr12.html>.

[HAMEL90] Prahalad, C., and Hamel, G. “*The core competence of the corporation*”, Harvard Bus. Rev. (May--June 1990), 79--91.

[QUINN94] Quinn, J.B., and Hilmer, “*F.G. Strategic outsourcing*”, Sloan Manage. Rev. (Summer 1994), 43--55.

[ERBER05] Georg Erber, Aida Sayed-Ahmed, “*Offshore Outsourcing A Global Shift in the Present IT Industry*”,Publicación Intereconomics, Editor Springer Berlin / Heidelberg, ISSN 0020-5346 (Print) 1613-964X (Online), Fascículo/ejemplar/número Volume 40, Number 2 / marzo de 2005, Categoría ARTICLES, DOI 10.1007/s10272-005-0141-8, Páginas 100-112, Subject Collection Empresas y Economía.

[GASCO06] Reyes Gonzalez, Jose Gasco, Juan Llopis, “*Information systems, Outsourcing, Benefits: Risk management*”, Journal: Industrial Management & Data Systems, Volume: 106, Number: 9, Year: 2006, ISSN: 0263-5577.

[CRAIG02] Craig R.D. and Jaskiel, S.P.: “*Systematic Software Testing*”, Artech House Publishers, Boston, (2002).

[BUMS03] Burnstein, I.: Practical “*Software Testing: A Process-oriented Approach*”. Springer Inc., New York (2003) ISBN 0-387-95131-8.

[PRESS05] Roger Pressman, “*Ingeniería del Software: Un Enfoque Práctico*”, Spanish, ISBN: 9701054733, 900 Páginas, 2005, Roger S. Pressman, McGraw-Hill, Sexta Edición.

[OREGAN02] G. O'Regan, “*A Practical Approach to Software Quality*”, Springer. (2002), ISBN 0-387-95321-3

[OWEN] Dawn M. Owens, Deepak Khazanchi, “*Software Quality Assurance*”, Handbook of Research on Technology Project Management, Planning, and Operations. ISBN: 9781605664002, Editor: Information Science Reference.

[SAIZ] José Manuel Sáiz Alvarez, Mónica García-Ochoa y Mayor, “*Externalización de servicios y alianzas estratégicas en la nueva economía del conocimiento*”, Economía industrial, ISSN

[SIKKA] “*Realizing the ROI of outsourced development*”, Vijay Sikka, pags. 135-141, <http://www.mityc.es/Publicaciones/Publicacionesperiodicas/EconomiaIndustrial/RevistaEconomiaIndustrial/370/135.pdf>

[SAIZ] Loh, L. y Venkatraman, N. (1992): “*Determinants of IT Outsourcing: a Cross-Sectional Analysis*”, Journal of Management Information Systems, vol. 9, núm. 1, pp. 7-24.

[QUILES07] Fernando J. Sánchez Quiles, “*Nueva tendencia en el aseguramiento de calidad de software identificada como la solución más efectiva para controlar desarrollos en modo outsourcing*”, Anales de mecánica y electricidad, ISSN 0003-2506, Vol. 84, Fasc. 5, 2007 (Ejemplar dedicado a: ciclo tecnología y cultura.

[PAN06] ”, Bin Xu; Xiao-Ping Pan; “*Optimizing Dual-Shore SQA Resource and Activities in Offshore Outsourced Software Projects* Electrical and Computer Engineering, 2006. CCECE '06. Canadian Conference on May 2006 Page(s):2405 – 2409, Digital Object Identifier 10.1109/CCECE.2006.277486

[ALAM] “*The project manager’s guide to software engineering best practices*”. Los Alamitos, CA: IEEE Computer Society.

[DAVIS99] Davis, W. S., & Yen, D. C. (1999). “*The information system consultant’s handbook: System analysis and design*”. Boca Raton, FL: CRC Press.

[GALIN04] Galin, D. (2004). “*Software quality assurance: From theory to implementation*”. Harlow, UK: Pearson Education Limited.

[SCOTT09] Scott, Mark. “*Outsourcing: Thriving at Home and Abroad.*” BusinessWeek Online (May 5, 2009): 15. Academic Search Premier, EBSCOhost.

[BROWN09] Alan S Brown, “*A shift in outsourcing offshore*”, Publication: Mechanical Engineering. New York: Mar 2009. Vol. 131, Iss. 3; pg. 24, 6 pgs, Source type: Periodical, ISSN: 00256501, ProQuest document ID: 1652498581

[CORB09] Michael F. Corbett Chairman, “*Outsourcing and the NEW Global Economy*”, International Association of Outsourcing Professionals (IAOP), Copyright © 2009 IAOP.

[SUESC00] Emilio Alvarez Suescun, María del Carmen Díaz Martín, “*Outsourcing de sistemas de información: factores determinantes*”, Localización: Actas del 7º Congreso de Economía Regional de Castilla y León : Soria, 23, 24 y 25 de noviembre de

2000, 2001 , pag. 371 Recoge los contenidos presentados a: Congreso de Economía Regional de Castilla y León (7. 2000. Soria).

[SOOD05] Sood, Robin. (2005). *IT, "Software and Services: Outsourcing and Offshoring"*. AiAiYo Books, LLC ©. ISBN:0976067439.

[IAOP] International Association Of Outsourcing (IAOP), "*The Outsourcing Life-cycle*" – 9 Stages, <https://www.outsourcingprofessional.org/firmbuilder/articles/34/200/945/default.asp>

[WALLACE06] Dolores.Wallace, Larry.Reeker, *C11 "Software_Quality"*, SCIRUS (Elsevier), Apr 2006, Technology Gaithersburg, Maryland 20899 USA, http://www.ppgia.pucpr.br/~alcides/Teaching/mestrado/FundamentosEngenhariaSoftware/artigos/C11-Software_Quality_Version_0_95.pdf

[BENTON08] Bruce Benton, "*Designing and Building a Software Test Organization*", Source ICST archive, Proceedings of the 2008 International Conference on Software Testing, Verification, and Validation , Pages: 414-422 ,Year of Publication: 2008, ISBN:978-0-7695-3127-4, Publisher IEEE Computer Society Washington, DC, USA.

[GREEN04] Jack Green_eld, Keith Short, Steve Cook, and Stuart Kent. "*Software Factories*". Wiley, Publising Inc., 2004.

[MKM] Editorial MKM, <http://www.mkm-pi.com/mkmpi.php?article5776>

[COMK] <http://www.telecomkh.com/es/comunicaciones-mpresariales/noticias/software/outsourcing/capgemini/cloud-computing/offshore/sogeti/2138>

[AIPMM10] <http://www.aipmm.com/html/newsletter/archives/000426.php> May 20, 2010. Business Value of Outsourced Product Testing By Somenath Nag.

[MACCALL78] Joseph P. Cavano, James A. McCall, "*A framework for the measurement of software quality*" , Pages: 133 – 139, Year of Publication: 1978, ACM Special Interest Group on Software Engineering.

[FUTR02] Robert T. Futrell; Donald F. Shafer; Linda I. Safer, "*Quality Software Project Management*", Publisher: Prentice HallPub Date: January 24, 2002, Print ISBN-10: 0-13-091297-2, Print ISBN-13: 978-0-13-091297-8

[KAN02] Stephen H. Kan, "*Metrics and Models in Software Quality Engineering*" Second Edition, Publisher: Addison-Wesley ProfessionalPub Date: September 16, 2002, Print ISBN-10: 0-201-72915-6, Print ISBN-13: 978-0-201-72915-3, Safari online books.

[TESTA09] Louis Testa, "*Growing Software*", 1st Edition, Publisher: No Starch PressPub Date: March 3, 2009, Print ISBN-13: 978-1-593-27183-1, Safari Online Books.

[GAFF] J. E. Gaffney, Jr. , "*Metrics in software quality assurance*", Pages: 126 - 130 , Year of Publication: 1981, ISBN:0-89791-049-4, ACM New York, NY, USA

[BOEHM76] Boehm, B. W., Brown, J. R., Lipow, M. L., “*Quantitative Evaluation of Software Quality*”. Proceedings of the 2nd International Conference on Software Engineering, San Francisco, California, United States, 1976, 592-605, IEEE Computer Society Press.

[ORTEGA02] Ortega, M., Pérez, M. & Rojas, T. (2002). “*A systemic quality model for evaluating software products*”. The 6th World Multiconference on Systemics, Cybernetics and Informatics SCI 2002 and The 8th International Conference on Information Systems, Analysis and Synthesis ISAS 2002, Orlando, Florida, USA. http://www.lisi.usb.ve/publicaciones/02%20calidad%20sistemica/calidad_24.pdf

[DROMEY95] Dromey, R. G., “*A model for software product quality*”. IEEE Transactions on Software Engineering 21, 1995, 146-162.

[SHAR08] Arun Sharma, Rajesh Kumar, P. S. Grover , “*Estimation of quality for software components: an empirical approach*”, Pages 1-10, Year of Publication: 2008, ISSN:0163-5948, ACM New York, NY, USA

[IEEE97] IEEE Estándar. 1028-1997, “*estándar de IEEE para las revisiones software*”.

[FAGAN76] Fagan, Michael E: “*Inspecciones del diseño y del código para reducir errores en el desarrollo de programa*”, *Diario de los sistemas de IBM*, Vol. 15, no. 3, 1976; “*Examinando diseños y código del software*”, *Proceso de datos automático*, Octubre de 1977; “*Avances en inspecciones del software*”, *Transacciones de IEEE en la tecnología de dotación lógica*, Vol. 12, no. 7, julio de 1986

[SOMM05] Ian Sommerville (Prentice Hall), “*Ingeniería del Software*”, Editorial: Pearson (2005, 7ª edición), Idioma: Español, ISBN: 8478290745, ISBN 13:9788478290741.

[LEVO00] Lévy-Levoyer, C. “*Gestión de las competencias. Cómo analizarlas, cómo evaluarlas, cómo desarrollarlas*”. Ed. Gestión 2000. Barcelona.1997.

[MUT05] Martínez Mut, B. “*Diseño de programas desde la perspectiva de los ECTS. Apuntes del Curso de formación permanente del mismo título*”. Instituto de Ciencias de la Educación. Universidad de Oviedo, Mayo de 2005.

[DEUSTO00] Universidad de Deusto y Universidad de Groningen. “*Tuning Educational Structures in Europe*”. Informe Final. Bilbao. 2000.

[OVD] Aquilino A. Juan, Javier de Andrés, Covadonga Nieto, Macamen Suárez, Juan Ramón Pérez, Agustín Cernuda, Candi Luengo, A. Belén Martínez, Miguel Riesco, Daniel F.Lanvín, Jose E. Labra, Marián D. Fondón, Jose Manuel Redondo, “*Definición de competencias específicas y genéricas del Ingeniero en Informática*”, Dpto. de Informática, Universidad de Oviedo.

[ISUS01] Echeverría, B., Isus, S. y Sarasola, L. “*Cualificaciones-Competencias: La contribución de los proyectos Leonardo Da Vinci y Adapt. Plan Nacional de Valoración*

(1995-1999)". Instituto Nacional de Empleo (INEM). Instituto Nacional de las Cualificaciones. Madrid, 2001.

[SEB03] Sebastián, A., Rodríguez, M.L. y Sánchez, M.F. "*Orientación Profesional. Un proceso a lo largo de la vida*". Ed. Dykinson, S.L. Madrid, 2003.

[CARRER] Career Space. "*Perfiles de capacidades profesionales genéricas de TIC. Capacidades profesionales futuras para el mundo del mañana*". Oficina de Publicaciones Oficiales de las Comunidades Europeas. Luxemburgo, 2001. www.carrer-space.com.

[ACM01] ACM-IEEE. Computing Curricula 2001. [www.computer.org/ education/c2001](http://www.computer.org/education/c2001).

[COPITI] COPIITI, Conferencia de la Profesión de Ingeniero e Ingeniero Técnico en Informática. "*Perfil de la profesión de Ingeniero en Informática y definición del currículo académico*". 2003.

[SEI99] CMU/SEI, "*A Software Engineering Body of Knowledge*" Version 1.0. Technical Report: CMU/SEI- 99-TR-004; ESC-TR-99-004. Abril 1999.

[TURLEY94] Richard T. Turley, James M. Bieman, "*Identifying essential competencies of software engineers*" Phoenix, Arizona, United States Pages: 271 – 278, Year of Publication: 1994, ISBN:0-89791-634-4, 2009 Department of Computer Science in the College of Natural Sciences at Colorado State University

[ANECA] "*Libro blanco Titulo de grado en ingeniería informática*", Agencia nacional de evaluación de la calidad y Acreditación (ANECA).

[ELC07] Miguel Ángel García Palomo, Mamdouh Elcuera, "*Capacitación de recursos testing*", RPM-AEMES, VOL. 4, N° Especial, Octubre 2007 ISSN: 1698-2029.

[DEUT79] Deutsch, M., "*Verification and Validation*", Software Engineering, R. Jensen y C.Tonies(eds.) Prentice Hall, 1979,pp. 329-408

[BEIZER90] Beizer, B., "*Software Testing Techniques*", 2ºed,"Van Nostrand Reinhold, 1990.

[GOOD75] John B. Goodenough, Susan L. "*Gerhart: Toward a Theory of Test Data Selection*". IEEE Trans. Software Eng. 1(2): 156-173 (1975)

[HETZEL88] Gelperin, D.; B. Hetzel (1988). "*The Growth of Software Testing*". *CACM* **31** (6). ISSN 0001-0782.

[IEEE98] IEEE Computer Society. "*IEEE 730-1998 Standard for Software Quality Assurance Plans*". IEEE Computer Society, (1998).

[DIJK72] Edsger W. Dijkstra." *Chapter I: Notes on structured programming*". páginas 1–82, 1972.

[TURING50] Turing, A. "*Checking a Large Routine*". Report of a Conference on High Speed Automatic Calculating Machines (Jan 1950), 67-69.

[BAKER57] Baker, C. Review of D.D.McCracken's "Digital Computer Programing". *Mathematical Tables and Other Aids to Computation* (oct 1957), 298-305.

[DEUST57] Deutsch.M.S, "*Software Verification and Validation*"; Realistic Project Approaches, Prentice-Hall, Englewood Cliffs, N.J. 1982.

[MILLER81] Miller, E., and Howden, W.E.Eds. "*Tutorial: Software Testing and Validation Techniques*". IEEE Computer Society Press, New York, 1981.

[DEUST] Deutsch.M.S, "*Software Project Verification and Validation*". Computer 14,4. 54-70.

[HOWDEN] Howden, W.E, "*Life-Cycle Software Validation*", Computer 14,4.

[NBS] "*Guidelines for Lifecycle Validation, Verification, and Testing of Computer software*", National Bureau of standards Reports NBS FIPS

[IEEE88] "*ANSI/IEEE STD 1008-1987 Standard for Software Unit Testing*". Institute of Electrical and Electronic Engineers, New York, 1988.

[HETZEL91] Hetzel, B., "*Software Testing: Some Troubling Issues and Opportunities*", BCS SpecialInterest Group in Software Testing, Dec 6, 1991.

[IEEE83] *ANSI/IEEE STD 829-1983 standard for Test Documentation*. Institute of Electrical and Electronic Engineers, New York, 1983.

[HART02] Hartman, A.: Is ISSTA "*Research Relevant to Industry*", ACM SIGSOFT Software Engineering Notes 27(4), 205-206 (2002).

[FUGG00] A. Fuggetta. Software Process: A Roadmap. 22nd International Conference on SoftwareEngineering (ICSE'2000), future of SoftwareEngineering Track, June 4-11, Limerick (Irlanda), ACM, 2000.

[LEOD90] R. McLeod Jr. "*Management Information Systems*". McMillan Publishing, New York, 1990.

[CANF04] Gerardo Canfora, Francisco Ruiz González, "Novática: *Revista de la Asociación de Técnicos de Informática*", ISSN 0211-2124, N°. 171, 2004 (Ejemplar dedicado a: Tecnología de proceso software) , pags. 5-8

[BOYER75] Boyer. R, Elpas. B, Levitt. K, 1975."A formal system for testing and debugging programs by symbolic execution". Not. 10, 6 tJune I, 234-245.

[CLARKE76] Clarke, L 1976, "A system to generate test data and symbolically execute programs". IEEE. Soft, 215-222.

[KOREL90] Korel., B 1990a, "Automated test data generation". *IEEE Trans. Softw. Eng.* 16, 8, 870-879.

[RAMA76] Ramamoorthy, C., Ho, S., AND CHEN, W. 1976, "On the automated generation of program test data". *IEEE Trans. Softw. Eng.* 2, 4, 293-300.

[BIRD82] Bird, D. and Muñoz, D., 1982. "Automatic generation of random self-checking test cases" *IBM Syst. J.* 22, 3, 229-245,

[INCE87] Ince, D., 1987, "The automatic generation of Testing Data" *Comput J.* 30, 1, 63-69.

[SHEW30] Walter Shewhart; "Economic Control of Quality of Manufactured Product"; Republished by American Society for Quality on December 1980; USA; 1930.

[DEMING86] W. Edward Deming; "Out of the Crisis"; Cambridge, MA: MIT Center for Advanced Engineering, 1986. (ISBN 0- 262-54115-7)

[IEEE04] "Guide to the Software Engineering Body of Knowledge" SWEBOK, 2004 version. IEEE Computer Society. <http://www.swebok.org>, 2004.

[PFLE01] Pfleeger S. "Software Engineering, 2nd Edition", ISBN: 0130290491, Prentice Hall, 2001.

[BOEHM79] B. W. Boehm. "Software Engineering; R&D Trends and Defense Needs". In R. Wegner, ed. Research. Directions in Software Technology. Cambridge, MA: MIT Press, 1979.

[IEEE90] IEEE/ANSI (1990), "IEEE Standard Glossary of Software Engineering Terminology, 610.12-1990".

[SPIRE] SPIRE Cases Studies. "Software Process Improvement in Regions of Europe". ESSI project 23873 SPIRE. (<http://www.cse.dcu.ie/spire/main.html>)

[GARCIA06] Javier García Guzmán, Antonio de Amescua Seco, Manuel Velasco de Diego. "Top 10 de problemas relativos a la mejora del proceso de verificación y validación en organizaciones intensivas en software". JISBD 2006: Jornadas de Ingeniería del Software y Bases de Datos 2006. October 2006. Barcelona, Spain.

[PEAR06] "Software Testing: Principles and Practices", ISBN: 9788177581218, Pearson Education India, 2006.

[BROOKS95] Nagappan, Murphy, B. Basili, V. "The influence of organizational structure on software quality", ISBN: 978-1-4244-4486-1 ISSN: 0270-5257 IDNumber: 10.1145/1368088.1368160, IEEE Xplore

[BROOKS95] F. P. Brooks, "The Mythical Man-Month", Anniversary Edition: Addison-Wesley Publishing Company, 1995.

[PAULK95] M. Paulk, C. Weber, B. Curtis, M. Chrissis, “*The Capability Maturity Model*”, Addison-Wesley, Reading MA, 1995.

[SUWAN96] Suwanassart, T., Carlson, R. Burnstein, I., “*Developing a Testing Maturity Model for software test process evaluation and improvement*”, Test Conference, 1996. Proceedings., International, ISBN: 0-7803-3541-4, IDNumber: 10.1109/TEST.1996.557106, IEEE Xplore .